

CASE FILE
COPY

NASA TECHNICAL
MEMORANDUM

NASA TM X-53997

SOME APPLICATIONS AND LIMITATIONS
OF THE FAST FOURIER TRANSFORM

By Edgar Hopper and Murl Newberry
Computation Laboratory

February 12, 1970

NASA

*George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama*

1. REPORT NO. TM X-53997	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.	
4. TITLE AND SUBTITLE Some Applications and Limitations of the Fast Fourier Transform		5. REPORT DATE February 12, 1970	
		6. PERFORMING ORGANIZATION CODE	
7. AUTHOR(S) Edgar Hopper and Murl Newberry		8. PERFORMING ORGANIZATION REPORT #	
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812		10. WORK UNIT NO.	
		11. CONTRACT OR GRANT NO.	
12. SPONSORING AGENCY NAME AND ADDRESS		13. TYPE OF REPORT & PERIOD COVERED Technical Memorandum	
		14. SPONSORING AGENCY CODE	
15. SUPPLEMENTARY NOTES Prepared by Computation Laboratory Science and Engineering Directorate			
16. ABSTRACT This report includes a discussion of power spectral density functions, autocorrelation functions, cross-spectral density functions, cross-correlation functions, and other related functions used in the analysis of a time series. The most direct route from a time series to a spectral density function is the Fourier transform. However, the computer time required to calculate a Fourier transform directly by conventional methods made this approach impractical. Recent developments in procedures for complex arithmetic have made the calculation of Fourier transforms of discrete time series feasible. The direct Fourier transform (DFT) and fast Fourier transform (FFT) techniques are evaluated. Examples of theoretical and practical problems are solved and the results are compared to the results from the conventional approach using STAN. Recommendations are made for quick-look analysis of structural vibrations.			
17. KEY WORDS STAN quick-look analysis FFT		18. DISTRIBUTION STATEMENT STAR Announcement <i>H. T. ...</i>	
19. SECURITY CLASSIF. (of this report) Unclassified	20. SECURITY CLASSIF. (of this page) Unclassified	21. NO. OF PAGES 65	22. PRICE \$3.00

ACKNOWLEDGMENT

The authors wish to express appreciation to Mr. Bill Chamblee for programming data used in this report.

TABLE OF CONTENTS

	Page
INTRODUCTION.	1
FOURIER TRANSFORM.	2
POWER SPECTRUM.	4
FILTERS OR WEIGHTS.	7
ACCURACY OF FFT.	11
EXPERIMENTAL RESULTS.	17
INTERPRETATION.	37
TYPICAL HARDWARE AND SOFTWARE CHARACTERISTICS	37
PROBLEM AREAS	43
CONCLUSION	43
APPENDIX: FAST FOURIER TRANSFORM	47
REFERENCES.	57
BIBLIOGRAPHY.	58

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Example of the convolution theorem	8
2.	FFT — Power spectral density, Channel 1, 1000 samples per second.	25
3.	FFT — Power spectral density, Channel 2, 1000 samples per second.	26
4.	FFT — Power spectral density, Channel 1, 1024 samples per second.	27
5.	FFT — Power spectral density, Channel 2, 1024 samples per second.	28
6.	STAN — Power spectral density, Channel 1, 1000 samples per second.	29
7.	STAN — Power spectral density, Channel 2, 1000 samples per second.	30
8.	STAN — Power spectral density, Channel 1, 1024 samples per second.	31
9.	STAN — Power spectral density, Channel 2, 1024 samples per second.	32
10.	Power spectral density from six-bit data, Channel 1	33
11.	Power spectral density from six-bit data, Channel 2	34
12.	Power spectral density from two-bit data, Channel 1	35
13.	Power spectral density from two-bit data, Channel 2	36
14.	Raw data, Channel 1.	44
15.	Raw data, Channel 2.	45

LIST OF TABLES

Table	Title	Page
1.	Spectral Analysis of $x(t) = 10 \cos 20.5\pi t$	14
2.	Square Wave.	16
3.	FFT — Time From 60.0 to 61.024	18
4.	FFT — Time From 60.0 to 61.0	18
5.	FFT — Time From 61.024 to 62.048	19
6.	FFT — Time From 61.0 to 62.0	19
7.	FFT — Time From 60.0 to 62.048	21
8.	FFT — Time From 60.0 to 62.0	21
9.	STAN — Time From 60.0 to 62.0.	22
10.	STAN — Time From 60.0 to 62.048	22
11.	Total Power For $100 e^{-10t}$	23
12.	Total Power For $100 e^{-2t} \sin 12 \pi t$	24
13.	Typical Characteristics of Environmental Tests	40
14.	FFT Timing Estimates.	41
15.	Typical FFA's and Their Basic Characteristics	42

SOME APPLICATIONS AND LIMITATIONS OF THE FAST FOURIER TRANSFORM

INTRODUCTION

The analysis of a time series is very important in many experiments (such as vibrational testing of structures). Seemingly, the two functions desired most often are the power spectral density function and the correlation function. Theoretically, if one is known exactly, so is the other. In practice, small errors in one may cause large errors in the other. This discussion includes power spectral density functions, autocorrelation functions, cross-spectral density functions, cross-correlation functions, and other related functions.

There are three separate aspects of the spectral analysis, (1) definition of spectrum, (2) calculation of the defined spectrum, and (3) interpretation of the calculated spectrum. All three problems are considered here.

The most direct route from a time series (discrete or continuous) to a spectral density function is the Fourier transform. In the past, the computer time required to calculate a Fourier transform made this approach impractical; as a result, most approaches were indirect. The correlation function for a limited number of lags was calculated, then the Fourier transform of the correlation function was calculated. Recent developments in procedures for complex arithmetic have made the calculation of Fourier transforms of discrete time series feasible. Because older methods have been in use longer, change to the direct Fourier transform method is slow.

This discussion explains the uses and limitations of the new approach using the Fast Fourier Transform.

FOURIER TRANSFORM

A detailed discussion of the Fourier transform, with the necessary and sufficient conditions for the existence of the transforms, is found in many textbooks. Here, the existence of the integrals or sums will be assumed to exist. In the Appendix, the procedure known as the Fast Fourier Transform is explained more fully.

If $x(t)$ is a time series and if the two integrals

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (1)$$

and

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (2)$$

exist, $X(f)$ is called the Fourier transform of $x(t)$, and $x(t)$ is called the inverse Fourier transform of $X(f)$. They, $X(f)$ and $x(t)$, are frequently called transform pairs.

In general, the time series that are recorded for analysis are finite and the existence of the preceding integrals are guaranteed by the assumption that the series is zero outside some time interval (usually $0 \leq t < T$). Under these assumptions these integrals become

$$X(f) = \int_0^T x(t) e^{-j2\pi ft} dt \quad (3)$$

$$x(t) = \int_{-F}^F X(f) e^{j2\pi ft} df, \quad (4)$$

where F is an upper bound of the frequencies found in $x(t)$. Also, in practice, the time series is a discrete time series (or must be made into one) of N values usually equally spaced. Therefore, the above integrals become

$$X(f) = (\Delta t) \sum_{i=0}^{N-1} x(t_i) e^{-j2\pi f t_i} \quad (5)$$

with

$$T = N \cdot \Delta t$$

and

$$x(t_k) = (\Delta f) \sum_{i=0}^{N-1} X(f_i) e^{+j2\pi f_i t_k}, \quad f_i = i(\Delta f), \quad (6)$$

where the last expression is summed on frequency. These sums are probably more familiar as

$$X(f) = (\Delta t) \sum x(t_i) \left[\cos 2\pi f t_i - j \sin 2\pi f t_i \right] \quad (7)$$

and

$$x(t) = (\Delta f) \sum X(f_i) \left[\cos 2\pi f_i t + j \sin 2\pi f_i t \right] \quad (8)$$

In the first formula, one may calculate $X(f)$ for any f by simply inserting that value of f in the sum and summing the values. In the second equation, the same is true of any t value if $X(f)$ is known for all f . These transforms are usually referred to as Discrete Fourier Transforms, or (DFT). If the sampling rate is sufficient, the value of the Discrete Fourier Transform should agree very closely with the Fourier transform evaluated at the same f value.

The Fast Fourier Transform (FFT) is just the DFT with some added restrictions and some innovations in the method of calculation. These restrictions are (1) N (the number of points) must be a composite number; (2) the calculated values of f are $0, \frac{1}{T}, \frac{2}{T}, \dots, \frac{N-1}{T}$; and (3) the

points must be equally spaced. For the limited values where $X(f)$ is calculated, the FFT and the DFT should agree to the accuracy of the program used. A great many of the programs available insist that $N = 2^a$ for some integer a . It seems to be desirable in all cases that N have no prime factor greater than 5.

Since the transform is the integral part of this study, it must be calculated with a reasonable degree of accuracy. However, the function of concern is the power spectrum and the discussion of accuracy will be included in that section.

POWER SPECTRUM

There are several definitions of power spectrum in the literature [1]. In theory, the direct definition of the power spectral density function of $x(t)$ is

$$S_x(f) = \lim_{T \rightarrow \infty} \frac{1}{2T} \left[\overline{X}_T(f) X_T(f) \right], \quad (9)$$

where

$$X_T(f) = \int_{-T}^T x(t) e^{-j2\pi ft} dt \quad (10)$$

and $\overline{X}_T(f)$ is the complex conjugate of $X_T(f)$. Since $x(t)$ is a finite series in the cases considered here, the definitions will be

$$S_X(f) = \frac{1}{T} \left[\overline{X}(f) X(f) \right] \quad (11a)$$

and

$$X(f) = \int_0^T X(t) e^{-j2\pi ft} dt \quad (11b)$$

This definition is for a time series where t runs from 0 to T and f runs from $-\frac{N}{2}\Delta f$ to $\frac{N}{2}\Delta f$ in the usual Discrete Fourier Transform approach, but f runs from 0 to $N(\Delta f)$ in the Fast Fourier Transform approach. Both of these are effectively two-sided spectral functions. It can be shown that if $X(f)$ is calculated from $-\frac{N}{2}\Delta f$ to $\frac{N}{2}\Delta f$, $X(k\Delta f) = \overline{X}(-k\Delta f)$ and if f is calculated at $0, \Delta f, \dots, N\Delta f$, then $X(k\Delta f) = \overline{X}[(N-k)\Delta f]$. Therefore in the first case, $S_x(f)$ is symmetric about $f = 0$, and in the second case it is symmetric about $\frac{N}{2}(\Delta f)$.

Some writers call the function of equations (11a) and (11b) the periodogram of $x(t)$ [2] and take m time records of length T and average these to get the spectral density function. Their definition would be

$$S_x(f) = \frac{1}{m} \sum_{i=1}^m S_{x_i}(f) \quad (12)$$

where $S_{x_i}(f)$ is the spectral density function of the i th time series by the previous definition or the periodogram by their definition. In actual problems, it is common practice to have a one-sided spectral function, defined as

$$G_x(f) = 2S_x(f), \quad 0 \leq f < \frac{N}{2}(\Delta f) \quad (13)$$

In this report, the spectral density function will be as defined by equations (11a) and (11b). If some modification is desired, it will be called the modified spectral function.

There are many ways to modify the spectral density function. One is given above where m series of length T are used and their spectral density functions are averaged. The more common modifications are to weight the time series before finding the Fourier transform, or by weighting the calculated Fourier transform before calculating the spectral density function. Usually, these procedures are referred to as weighting, smoothing or filtering. Some of the common weighting functions or filters are given later in this report [1, 2, 3]. However, for deterministic

functions, the results are more accurate without the filtering when the time T and the sampling rate are correctly chosen.

Some other useful functions, easily calculated from the FFT and power spectral density functions, are discussed here. If $x(t)$ and $y(t)$ are two time series of the same length and sample rate, the cross power spectral density function is

$$S_{xy}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} \{ \overline{X}(f) Y(f) \} \quad , \quad (14)$$

or for our approximation

$$\hat{S}_{xy}(f) = \frac{1}{T} \{ \overline{X}(f) Y(f) \} \quad (15)$$

Also, the autocorrelation function and cross-correlation function are Fourier transforms of the power spectral density function and cross spectral density function, respectively, i.e.,

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau} df \quad (16)$$

and

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} S_{xy}(f) e^{j2\pi f\tau} df \quad (17)$$

Under the restrictions of the Appendix, if $S_x(f)$ is entered instead of the time series in the FFT routine, the autocorrelation function is the output; if $S_{xy}(f)$ is used for input, the cross-correlation function is the output.

Another function frequently desired is the phase angle between two time series. If the cross power spectral density function $S_{xy}(f)$ is written as a complex function

$$S_{xy}(f) = C_{xy}(f) - jQ_{xy}(f) \quad , \quad (18)$$

then the phase angle θ is

$$\theta = \tan^{-1} \frac{Q_{xy}(f)}{C_{xy}(f)} \quad (19)$$

If the correlation functions are desired, it is recommended by some [1, 2] that the time series $x(t)$ or $y(t)$ be altered by addition of the same number of zeros as the number of values of the correlation function desired. For example, if the time series consists of 1024 points and 100 values of the correlation function are desired, the last 100 values would be replaced by zeros. With the altered time series, the correlation functions computed via the FFT yields better agreement to the direct correlation method.

FILTERS OR WEIGHTS

Two commonly used programs for vibration analysis are RAVAN [4] and STAN¹. These use the Blackman-Tuckey method.

In the Blackman-Tuckey method of calculating the power spectra, filtering is necessary even for periodic deterministic data. With the FFT approach, filters are not required for periodic data when the length of the record is chosen correctly.

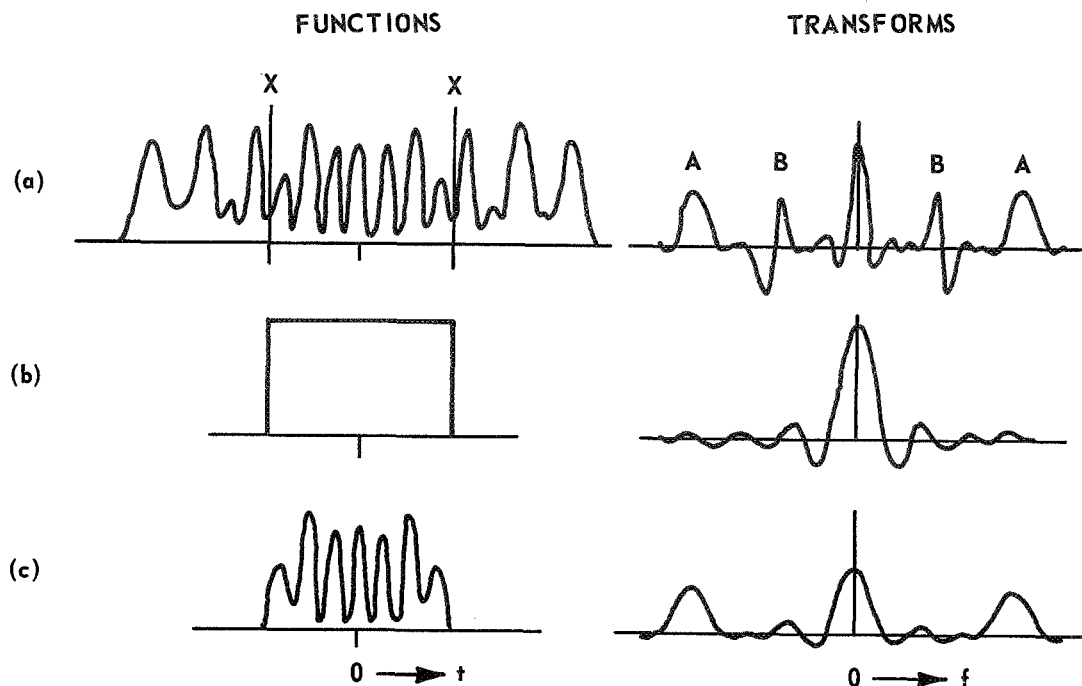
The filter or weight should be designed for the particular problem. However, some of the common filters are discussed here.

Defining the time series $x(t)$ to be zero outside a finite interval $-T \leq t < T$ is equivalent to using a weight function $w(t) = 1$ on this interval and zero elsewhere. Now, according to Fourier transform theory, the Fourier transform of a product $w(t) x(t)$ is the convolution of $W(f)$ with $X(f)$. Mathematically, this can be expressed as the transform of $w(f) x(t)$:

$$\int_{-\infty}^{\infty} X(f_0) W(f - f_0) df_0 \quad (20)$$

The effect of using a short sample of the record is shown in Figure 1 [5].

1. Newberry, M. H.: The Statistical Analyzer Program (STAN). Internal Note IN-COMP-67-1, George C. Marshall Space Flight Center, September 1967.



NOTE: Ignoring the early and late parts of the function has the effect of "blurring" its spectrum. As shown here, the broad peaks AA of the transform are little affected; the rapid oscillations BB almost disappear; and the narrow peak at zero frequency is reduced. This conclusion is physically reasonable since cutting off the ends of the function (a) leaves most of the high frequency oscillations and simultaneously deletes the slow ones and halves the area under the curve. The function has been chosen symmetrically to avoid the need to represent complex values in the transform.

Figure 1. Example of the convolution theorem.

If the first line (a) is $x(t)$ and its transform $X(f)$, and the second line (b) is the weight function $w(t)$ and its transform $W(f)$, then the last line (c) is $w(t)x(t)$ and its transform. Notice that the higher frequency components of the transforms of $x(t)$ and $w(t)x(t)$ are very similar. However, in this case, the lower frequency components are greatly altered. Also, the value at $f = 0$ is much lower in the weighted case. The process of convolution causes a cancellation when there is a rapid oscillation, as in this case.

For the Discrete Fourier Transform or Fast Fourier Transform, this same situation applies unless the frequencies are discrete and the time interval is chosen so that the calculated values are the actual frequencies present. In this case with deterministic functions, the results are very accurate.

This problem of using a finite record cannot be avoided. Comparison of this analysis with that of a longer segment of the time series will often help one to see the actual situation since the narrower bandwidth tends to separate the peaks.

One of the weights recommended is [2]

$$\begin{aligned}
 w(t) &= \frac{1}{2} \left(1 - \cos \frac{\pi t}{0.1T} \right) \text{ for } 0 \leq t < 0.1T, \\
 w(t) &= 1 \text{ for } 0.1T \leq t \leq 0.9T, \\
 w(t) &= \frac{1}{2} \left[1 + \cos \pi \frac{(t - 0.9T)}{0.1T} \right] \text{ for } 0.9T < t \leq T.
 \end{aligned} \tag{21}$$

If one considers the above weight function and compares its spectra or transform with the square weighting function above, one sees that the side lobes are much smaller. This means that the frequencies in these side bands contribute less to the value calculated for a given frequency [1].

Another weight function used [6] is

$$w(t_j) = \sin^2 \frac{\pi j}{N} \tag{22}$$

or x_j is weighted by $w(t_j)$. One of the advantages of this filter is that the Fourier coefficients a_k 's and b_k 's can be calculated without the weight function, and then the coefficients can be weighted so that

$$a'_k = \frac{1}{2} \left(-a_{k-1} + 2a_k - a_{k+1} \right) \tag{23}$$

and

$$b'_k = \frac{1}{2} \left(-b_{k-1} + 2b_k - b_{k+1} \right) \quad (24)$$

The document cited in Reference 6 recommends further smoothing by calculating

$$P_k = \frac{1}{2} \left(a_k'^2 + b_k'^2 \right) , \quad (25)$$

where P_k is the average power due to the k th frequency component, and then calculating

$$P'_k = \frac{1}{4} \left(P_{k-1} + 2P_k + P_{k+1} \right) \quad (26)$$

when noise is present. This last step is equivalent to weighting the auto-covariance function with $\cos^2 \frac{\pi\tau}{n}$ in the STAN program.

There are many other documented weighting or filtering approaches [2, 5, 6].

It should be realized that smoothing, filtering or weighting basically degrades the resolution. Sharp peaks that might occur in a particular record are smoothed out so that the analysis of different records of the same general happening are more nearly the same. The improvement is in stability or reproducibility. The user should decide first whether he wants resolution or stability. Then, if he desires stability, the proper filter should be chosen for his problem.

One of the most commonly used class of weights is a generalized class of cosine tapers. The transform of the generalized form is

$$W(2\pi f, m) = \left[\frac{\sin\left(\frac{2m-1}{2m}\right) \pi f T}{\left(\frac{2m-1}{2m}\right) \pi f} \right] \cdot \left[\frac{\left(\frac{2\pi m}{T}\right)^2}{\left(\frac{2\pi m}{T}\right)^2 - 4\pi^2 f^2} \right] \cdot \left[\cos \frac{\pi f T}{2m} \right], \quad (27)$$

where $m = fT$. This weight is capable [3] of reducing the contribution from adjacent frequencies separated by more than $f = \frac{m}{T}$. However, optimum choice of m requires the advance knowledge of the minimum separation of any two frequencies of interest.

ACCURACY OF FFT

For the FFT problem, the main sources of error are sampling rate, numerical integration, quantization and the number of calculations required.

The numerical integration scheme approximates the Fourier integral by a series

$$\int_0^T x(t) (\cos 2\pi ft - j \sin 2\pi ft) dt \cong (\Delta t) \left[y_0 + y_1 \dots + y_{N-1} \right] \quad (28)$$

where $N \cdot \Delta t = T$ and

$$y_i = x(t_i) \left[\cos 2\pi ft_i - j \sin 2\pi ft_i \right] \quad (29)$$

This particular form does not lend itself to easy theoretical analysis. However, it is very similar to the trapezoidal integration formula

$$(\Delta t) \left[\frac{y_0}{2} + y_1 + y_2 \dots + y_{N-1} + \frac{y_N}{2} \right], \quad (30)$$

for which the error estimates are known. The upper bound to the error in this formula is [7]

$$E = \max \frac{Ny''(\Delta t)^3}{12} \quad (31)$$

where y'' is the second derivative of $x(t)$ $[\cos 2\pi ft - j \sin 2\pi ft]$. If it is assumed that $x(t)$ can be expanded into a trigonometric series, the four terms involved for each f_i are $(\cos 2\pi f_i t) (\cos 2\pi ft)$, $-j(\sin 2\pi f_i t) (\sin 2\pi ft)$, $-j(\cos 2\pi f_i t) \sin 2\pi ft$, $\sin 2\pi f_i t \cos 2\pi ft$.

These terms are very similar and only one is treated here. Since the integrals of these terms vanish unless $f = f_i$, consider $y(t) = \cos^2 2\pi ft$.

Then, $y'(t)$ is $-4\pi f \cos 2\pi ft (\sin 2\pi ft)$, $y''(t) = 8\pi^2 f^2 (\sin^2 2\pi ft - \cos^2 2\pi ft)$, and $|y''(t)| \leq 8\pi^2 f^2$. Now, let $T = 1$ unit of time. Then, if k samples are taken on each cycle of $y(t)$, $N = kf$ and $\Delta t = \frac{1}{kf}$.

Hence,

$$E \leq \frac{(kf) (8\pi^2 f^2) \left(\frac{1}{kf}\right)^3}{0.12} = \frac{8\pi^2}{12k^2}$$

The integral of $\cos^2 2\pi ft$ between 0 and 1 is $\frac{1}{2}$. Hence, the maximum

percentage error is $\frac{16\pi^2}{12k^2}$ or roughly $\frac{14}{k^2}$. This indicates the maximum error for 10 samples per cycle of the given frequency is 14 percent. In deterministic cases, the error is much smaller than the theoretical limit when the sample rate exceeds 5 samples per cycle. For lower rates, it is very close. In one problem with 3 samples per cycle, the error was 100 percent. Thus, errors resulting from sample rate and errors resulting from numerical integration procedure are closely associated.

The quantization errors must be treated statistically. If it is assumed that the quantization errors satisfy certain common statistical models, the errors in the final answers are very small. Two examples that will be discussed later indicate that this conclusion is valid in these cases.

The propagation and round-off errors have been treated by Welch [8] and have not been treated here. Using his assumptions, the percentage error was very low, usually less than 0.1 percent.

While error estimates are very worthwhile, experiments with functions where the answers are known or can be compared with some accepted standard are very desirable.

The first case chosen was the deterministic function $X(f) = 10 \cos 20.5 \pi t$, ($f = 10.25$). The objective in this experiment was two-fold: (1) to show the effect of the record length T on resolution, (2) to show the accuracy of the method.

The results when the time T was $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, 1, 2, and 4 seconds are shown in Table 1. It is obvious if one graphs the results that as T increases the spectral density function tends toward a peak at 10 1/4 cycles per unit of time. At $T = 4$, the function is zero everywhere except at 10 1/4 cycles as it should be. Moreover, the peak value is 50 and this is the theoretical value. For T , any integral multiple of 4, the results will repeat.

The values for the power spectral density at other frequencies when $T < 4$ are due to the model. Effectively the FFT calculates the Fourier series coefficients for a function which agrees with the given function from 0 to T , and then repeats itself with period T . Clearly, in the given case, the two functions were not the same until $T = 4$. These values are usually referred to as the results of leakage.

Since nothing is known about the values of the spectral density function between calculated values, discretion must be used in interpreting a single analyzed record. If a frequency component falls between calculated values of the spectral density function, the calculated values on each side will compensate for the missed frequency. The Fourier coefficients decay as $\frac{1}{f_a - f_c}$, where f_a is the actual frequency and f_c is the calculated frequency.

Studying two or more of these cases with $T < 4$ should give a better understanding than one case with a fixed T .

The second example chosen was a square wave with amplitude 64 and at 32 cycles per second. The sample rate was chosen to be 2048 samples

TABLE 1. SPECTRAL ANALYSIS OF $x(t) = 10 \cos 20.5\pi t$

Frequency	N = 8; T = 1/8		N = 16; T = 1/4		N = 32; T = 1/2		N = 64; t = 1		N = 128; t = 2		N = 256; t = 4	
	G(f) (Unit) ²	PSD Unit ² /Hz	G(f) (Unit) ²	PSD Unit ² /Hz	G(f) (Unit) ²	PSD Unit ² /Hz	G(f) (Unit) ²	PSD Unit ² /Hz	G(f)	PSD	G(f) (Unit) ²	PSD Unit ² /Hz
0.000 0.250 0.500 0.750	0.865	6.924	0.0737	0.2948	0.0607	0.1215	0.0968	0.0968	0.0244	0.0122		0.0000
1.000 1.250 1.500 1.750							0.0987	0.0987	0.0255	0.0127		
2.000 2.250 2.500 2.750					0.0655	0.1310	0.1046	0.1046	0.0290	0.0145		
3.000 3.250 3.500 3.750							0.1158	0.1158	0.0359	0.0180		
4.000 4.250 4.500 4.750			0.2333	0.9332	0.0837	0.1674	0.1346	0.1346	0.0480	0.0240		
5.000 5.250 5.500 5.750							0.1665	0.1665	0.0694	0.0347		
6.000 6.250 6.500 6.750					0.1373	0.2748	0.2237	0.2237	0.1101	0.0551		
7.000 7.250 7.500 7.750							0.3393	0.3393	0.1978	0.0989		
8.000 8.250 8.500 8.750	5.881	47.049	3.0507	12.203	0.3812	0.7625	0.6322	0.6322	0.4365	0.2183		
9.000 9.250 9.500 9.750							1.839	1.839	1.5010	0.7505		
10.000 10.250 10.500 10.750					24.4255	48.8511	41.541	41.541	39.9042	19.952	200.0	50.0000
11.000 11.250 11.500 11.750							4.1890	4.1890	4.7187	2.359		
12.000 12.250 12.500 12.750			7.4488	29.7	0.3986	0.7972	0.7014	0.7014	0.9226	0.4613		0.0000

TABLE 1. (Concluded)

Frequency	N = 8; T = 1/8		N = 16; T = 1/4		N = 32; T = 1/2		N = 64; t = 1		N = 128; t = 2		N = 256; t = 4	
	G(f) (Unit) ²	PSD Unit ² /Hz	G(f) (Unit) ²	PSD Unit ² /Hz	G(f) (Unit) ²	PSD Unit ² /Hz	G(f) (Unit) ²	PSD Unit ² /Hz	G(f)	PSD	G(f) (Unit) ²	PSD Unit ² /Hz
13.000 13.250 13.500 13.750							0.2600	0.2600	0.3977 0.2938	0.1988 0.1469		0.0000
14.000 14.250 14.500 14.750					0.0698	0.1396	0.1285	0.1285	0.2276 0.1828	0.1138 0.0914		
15.000 15.250 15.500 15.750							0.0739	0.0739	0.1510 0.1275	0.0755 0.0638		
16.000 16.250 16.500 16.750	0.4995	3.996	0.9435	3.774	0.0238	0.0477	0.0467	0.0467	0.1097 0.0958	0.0548 0.0479		
17.000 17.250 17.500 17.750							0.0315	0.0315	0.0847 0.0758	0.0424 0.0379		
18.000 18.250 18.500 18.750					0.0105	0.0210	0.0223	0.0223	0.0684 0.0623	0.0342 0.0312		
19.000 19.250 19.500 19.750							0.0164	0.0164	0.0572 0.0528	0.0286 0.0264		
20.000 20.250 20.500 20.750			0.4362	1.7446	0.0052	0.0105	0.0124	0.0124	0.0491 0.0459	0.0245 0.0229		
21.000 21.250 21.500 21.750							0.0096	0.0096	0.0431 0.0407	0.0216 0.0204		
22.000 22.250 22.500 22.750					0.0028	0.0056	0.0077	0.0077	0.0386 0.0367	0.0193 0.0184		
23.000 23.250 23.500 23.750							0.0063	0.0063	0.0350 0.0336	0.0175 0.0168		
24.000	0.0724	0.5794	0.2915	1.6604	0.0015	0.0030	0.0052	0.0052		0.0162		
25.000							0.0044	0.0044		0.0151		
26.000							0.0038	0.0038		0.0142		
27.000							0.0033	0.0033		0.0135		
28.000			0.237	0.9487	0.0005	0.0009	0.0030	0.0030		0.0130		0.0000
TOTALS				50.85		51.33						

per second with $T = 1$ second. The objective was to compare the accuracy of the FFT with the theoretical answers and also to compare it with the results from RAVAN and STAN for a similar time series.

The calculated values of the PSD and the theoretical values for the first 10 components are tabulated in Table 2. The percentage error based on the maximum theoretical value is roughly 0.04 percent. The actual value of the error decreases with higher frequency.

A direct comparison with RAVAN and STAN values was not made since the example recorded in STAN² used a square wave of 100 cycles per second and 8000 samples per second. However, the actual errors in these cases increased with increasing frequency. With the FFT, the percentage error fell between the values of RAVAN and STAN for the lower frequencies. For higher frequencies, the FFT answers had smaller percentage errors than either RAVAN or STAN, even though their sample rate was higher per cycle.

TABLE 2. SQUARE WAVE (Amplitude 64)

Freq.	FFT (PSD)	Freq.	Theory (PSD)
32	3318.76	32	3320.0925
96	367.577	96	368.899
160	131.577	160	132.8037
224	66.487	224	67.757
288	39.763	288	40.989
352	26.268	352	27.439
416	18.544	416	19.646
480	13.739	480	14.756
544	10.572	544	11.488
608	8.400	608	9.197

2. *ibid.*

EXPERIMENTAL RESULTS

Two channels of data from test AS-506 were used for several different studies. The first four of these studies were to show the effect of the sampling rate and the part of the records used. All four were taken from the two channels of data with a time slice from 60 seconds to 62.048 seconds.

The first analysis used the first 1.024 seconds of data from each channel with a sampling rate of 1000 samples per second. The predominant (largest PSD) frequencies and their magnitudes are shown in Table 3 for both channels.

The second analysis used a time slice of 1 second from 60.0 to 61.0 seconds of each channel. The sample rate was 1024 samples per second. The results of this analysis are shown in Table 4.

Even though the sample rate and time interval were very nearly the same, some changes in the predominant frequencies did occur. No rigorous explanation of these differences is offered. Since the different lengths of record forced the calculation of the FFT at different discrete values of frequency, these variations may result from the sharpness of the peaks on the curves.

The third and fourth analyses were made to further check the effect of sampling rate and length of record and also to check whether these particular time series were self-stationary with the given lengths of record.

If the time series were self-stationary with time T , the results from any two segments of the record with length T would be the same.

Table 5 shows the results for a time slice from 61.024 to 62.048 seconds with a sample rate of 1000 per second. Table 6 shows the results of a time slice from 61.0 to 62.0 seconds with a sample rate of 1024 per second.

Again, there is considerable variation between Tables 5 and 6 because of the sample rates and lengths of record. Also, Tables 3 and 5 differ as do Tables 4 and 6. This indicates that the time series are not self-stationary with records of length 1 second or 1.024 seconds. It should be noted that the predominant frequency is basically the same for the given channel in all four cases. Many of the others occur in each case but their rankings are different.

TABLE 3. FFT — TIME FROM 60.0 TO 61.024

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
186.523	5.513	187.500	26.783
156.250	2.690	281.250	4.537
177.734	2.172	186.523	4.263
150.391	2.094	337.891	3.973
145.508	2.042	0.977	3.915
151.367	1.806	213.867	3.834
168.945	1.666	235.352	3.065
187.500	1.513	182.617	2.979
162.109	1.474	228.516	2.675
159.180	1.419	207.031	2.662

TABLE 4. FFT — TIME FROM 60.0 TO 61.0

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
187.00	8.015	188.00	28.641
153.00	2.729	282.00	5.52
156.00	2.262	1.00	5.157
202.00	1.831	2.00	4.924
152.00	1.726	344.00	4.173
178.00	1.672	226.00	3.643
163.00	1.662	215.00	3.622
6.00	1.4914	219.00	3.315
1.00	1.424	183.00	2.939
169.00	1.135	365.00	2.632

TABLE 5. FFT — TIME FROM 61.024 TO 62.048

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
186.523	10.208	187.50	21.869
0.977	6.998	188.477	8.372
158.203	3.483	2.930	6.717
170.898	2.668	215.820	4.602
163.086	1.768	226.562	3.817
10.953	1.734	204.102	3.774
147.461	1.367	220.703	3.541
180.664	1.312	239.258	3.428
195.312	1.258	214.844	3.378
152.344	1.150	347.656	3.269

TABLE 6. FFT — TIME FROM 61.0 TO 62.0

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
187.00	5.838	188.00	23.727
186.00	4.601	189.00	15.633
160.00	1.965	215.00	8.054
194.00	1.443	282.00	5.739
7.00	1.339	281.00	4.687
159.00	1.286	233.00	4.063
154.00	1.285	249.00	3.714
60.00	1.278	205.00	3.655
180.00	1.263	172.00	3.579
140.00	1.235	376.00	3.450

The next two analyses were made to compare the results of runs of length $2T$ with those of length T and also to compare the results of the same analyses performed by STAN.

*

Table 7 shows the results of the analysis of 2.048 seconds of data with a sampling rate of 1000 per second and Table 8 shows the results of analysing a 2 second record with sampling rate of 1024 samples per second. The agreement between these two cases seems to be better than for the shorter records.

Tables 9 and 10 show the results of analysing the same data as Tables 7 and 8 but using STAN. The STAN program was set up to find the predominant frequencies up to 200 cycles per second. There is very good agreement between these methods as to the predominant frequency in each case. It should be noted that the slightly different sampling rate and length of records affected the results of the STAN program fully as much as it did the FFT.

These results are also shown in Figures 2 through 9. Figure 2 should be compared with Table 7, channel 1, and Figure 3 to Table 7, channel 2. Figure 4 corresponds to Table 8, channel 2. Figure 6 corresponds to Table 9, channel 1, and Figure 7 to Table 9, channel 2. Figures 8 and 9 correspond to Table 10, channels 1 and 2, respectively.

Two other experiments were performed to study the effect of the number of bits. When the data was changed from 10 bits to 6 bits, the change in the larger peaks was only about 10 percent. Changing from 10 bits to 2 bits gave about the same results for the major peak. These results are shown in Figures 10 through 13. From this, it appears that the number of bits used is not too significant.

Two further analyses were performed. The purpose of these two was to check the results of the FFT when used to calculate the frequency response function of a system. Two time series, $100 e^{-10t}$ and $100 e^{-2t} \sin 12 \pi t$, were chosen. Since these functions are obviously not stationary, the spectral density function was modified. In this study, the quantity $2 \overline{X(f) X(f)}$ was used and compared with the theoretical value of $2 |H(f)|^2$ where $H(f)$ is the frequency response corresponding to the given time response. The factor of 2 was included in order to use only positive frequencies. These results are given in Table 11 for $100 e^{-10t}$ and Table 12 for $100 e^{-2t} \sin 12 \pi t$. The totals shown in the tables are for the integrals

TABLE 7. FFT — TIME FROM 60.0 TO 62.048

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
186.523	6.961	187.500	13.431
0.488	4.135	187.958	7.586
169.434	1.474	188.477	4.009
1.465	1.406	188.956	3.576
0.977	1.402	215.332	2.904
155.762	1.294	0.488	2.820
152.832	1.181	281.250	2.675
187.012	1.1001	223.145	2.549
4.395	1.063	216.306	2.064
60.059	0.967	209.961	1.997

TABLE 8. FFT — TIME FROM 60.0 TO 62.0

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
187.00	3.764	188.0	13.898
186.5	3.457	187.5	7.615
0.500	2.023	189.0	6.251
187.5	1.635	0.50	5.301
60.0	1.568	281.5	4.505
156.0	1.351	188.5	3.162
169.5	1.140	189.5	2.352
177.5	1.049	212.5	2.236
152.0	1.034	216.5	2.230
194.0	0.944	206.0	2.156

TABLE 9. STAN — TIME FROM 60.0 TO 62.0

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
187.0	10.005	188.0	30.393
156.0	2.434	182.0	3.122
170.0	2.105	196.0	2.621
6.0	2.000	193.0	2.313
153.0	1.978	165.0	2.042
163.0	1.841	8.0	1.825
159.0	1.772	172.0	1.628
178.0	1.595	176.0	1.595
60.0	1.533	60.0	1.357
195.0	1.343	155.0	1.350

TABLE 10. STAN — TIME FROM 60.0 TO 62.048

Channel 1		Channel 2	
Freq.	Amp. (PSD)	Freq.	Amp. (PSD)
186.8	8.898	187.8	26.646
155.8	1.753	3.0	3.240
152.3	1.555	1.0	2.615
169.3	1.407	182.3	2.235
177.3	1.399	180.8	1.820
170.8	1.369	192.3	1.713
4.5	1.313	195.3	1.711
163.3	1.151	193.8	1.402
158.3	1.036	184.8	1.266
60.1	1.022	171.8	1.203

TABLE 11. TOTAL POWER FOR $100 e^{-10t}$

Freq.	1000 Samples/Sec 1.024 sec	Theory
0	104.17	100.00
0.977	146.68	145.30
1.953	80.57	79.81
2.930	46.01	45.58
3.906	28.75	28.47
4.883	19.39	19.21
5.859	13.87	13.74
6.836	10.38	10.28
7.813	8.05	7.97
8.789	6.41	6.35
9.766	5.23	5.17
Total	497.86	500.00

of the area under the curves $2 \overline{X(f)} X(f)$ and $2 |H(f)|^2$, respectively. It should be noted that the error in $2 |H(f)|^2$ is $4 |H(f)|$ times as large as the error in $|H(f)|$.

Hence, for the value of $|H(f)|$ at 6 cycles per second in Table 12, the error should be approximately 1 part in 50 for 2 seconds of data and about 1 part in 500 000 when 16 seconds of data were used.

TABLE 12. TOTAL POWER FOR $100 e^{-2t} \sin 12\pi t$

Freq.	512 Samples/ Sec 2 sec	512 Samples/ Sec 4 sec	256 Samples/ Sec 4 sec	512 Samples/ Sec 16 sec	Theory
0	6.7392	6.9865	6.9675	6.9865	6.9968
1	14.2477	14.7744	14.7353	14.7843	14.7972
2	17.0152	17.6442	17.6016	17.6561	17.6701
3	23.7952	24.6747	24.6245	24.6914	24.7082
4	42.7892	44.3709	44.3037	44.4006	44.4642
5	131.7444	136.6147	136.5003	136.7063	136.7443
6	1203.7800	1248.2800	1248.2704	1249.1187	1249.1188
7	94.4052	97.8950	97.9904	97.9607	97.9291
8	21.8684	22.6768	22.7245	22.6920	22.6760
9	8.5886	8.9061	8.9363	8.9121	8.9020
10	4.2691	4.4269	4.4484	4.4299	4.4228
11	2.4271	2.5169	2.5330	2.5186	2.5132
12	1.5062	1.5619	1.5747	1.5630	1.5587
Total Power	1242.70	1244.74	1244.74	1246.03	1246.49

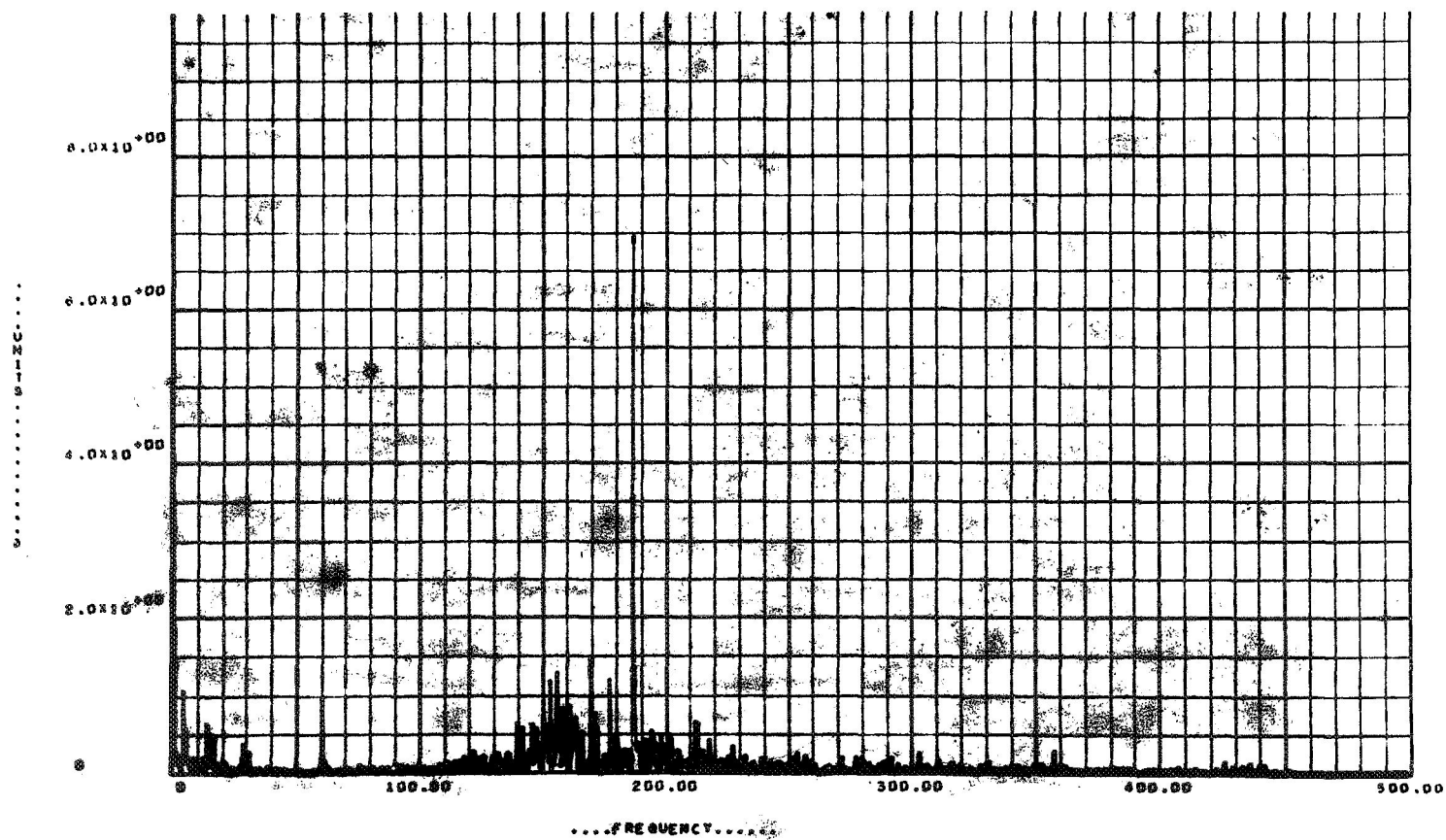


Figure 2. FFT — Power spectral density, Channel 1, 1000 samples per second.

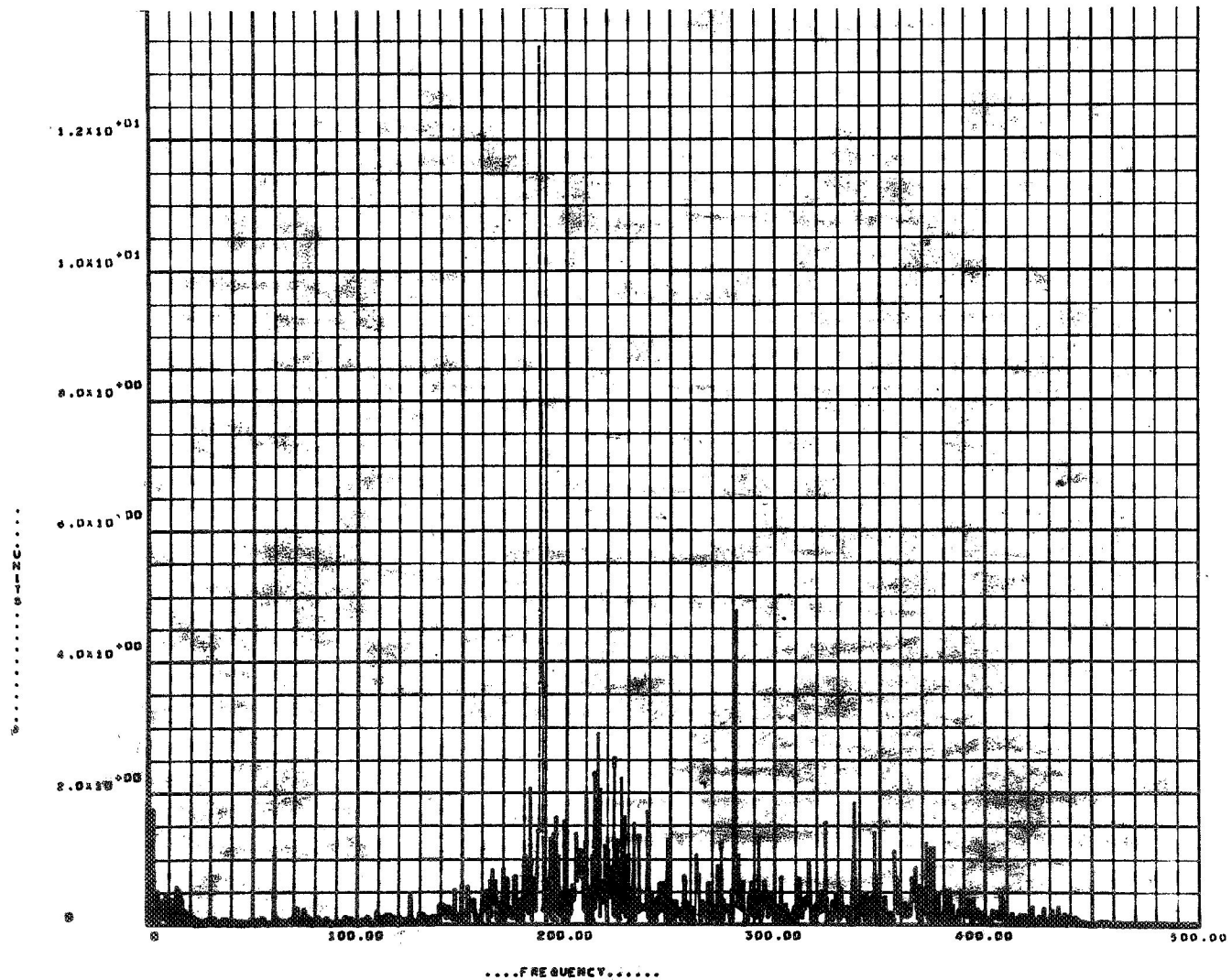


Figure 3. FFT — Power spectral density, Channel 2, 1000 samples per second.

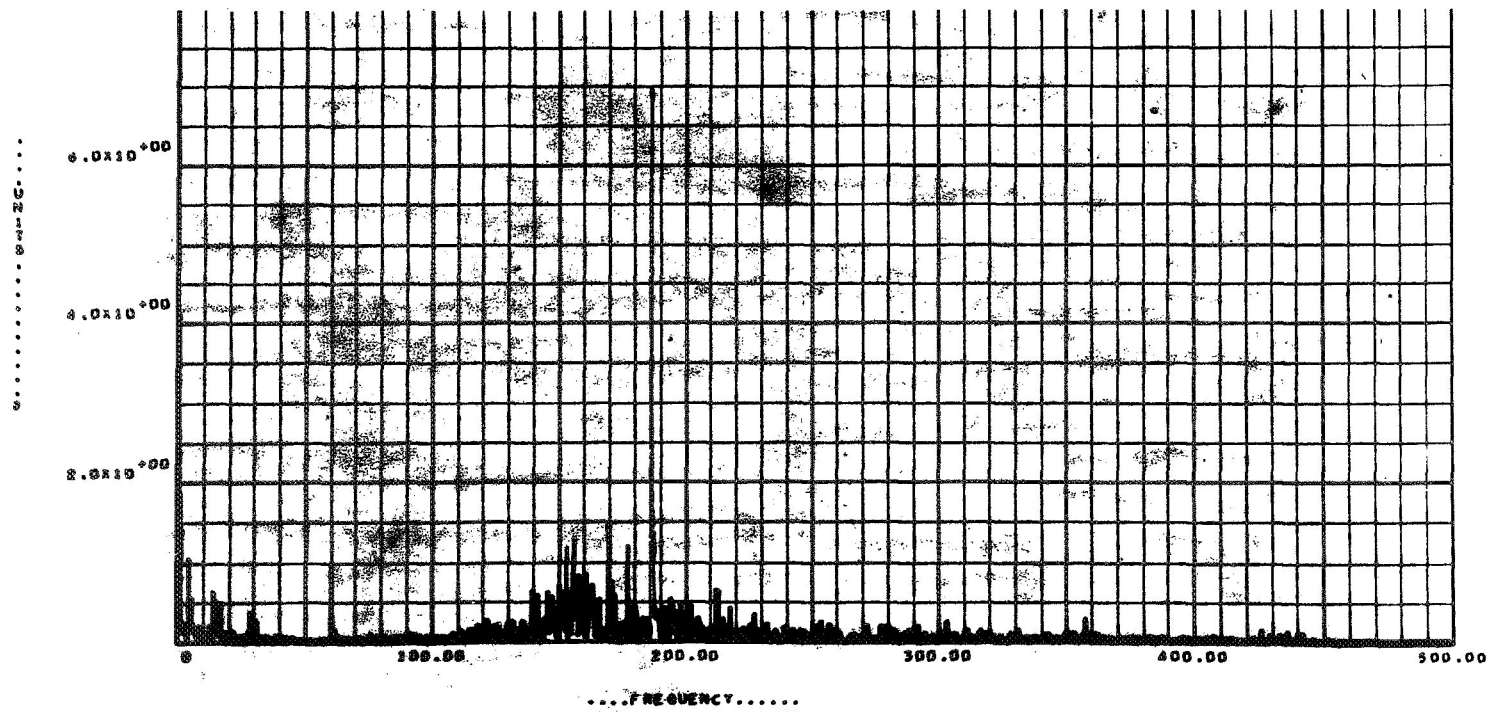


Figure 4. FFT — Power spectral density, Channel 1, 1024 samples per second.

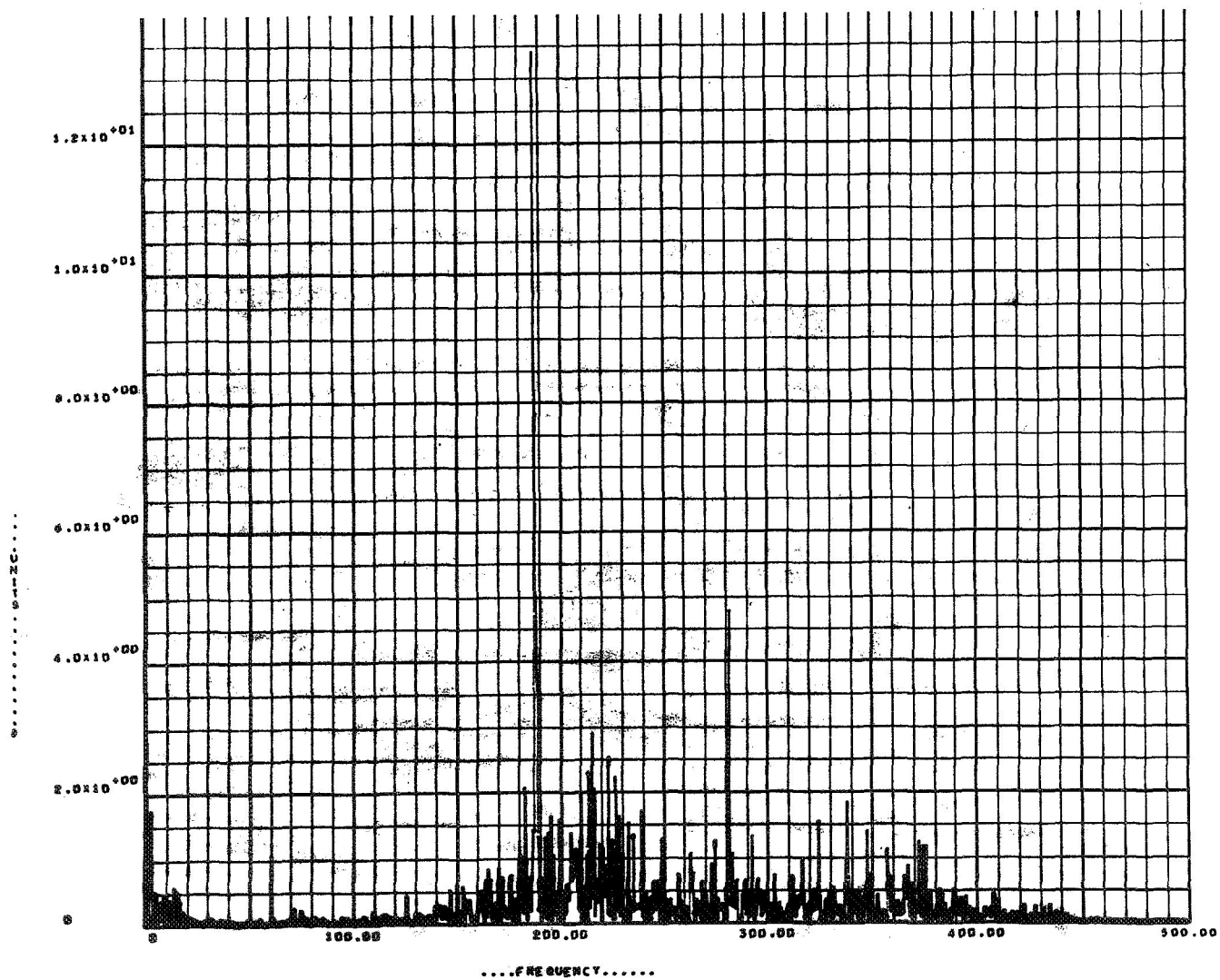


Figure 5. FFT — Power spectral density, Channel 2, 1024 samples per second.

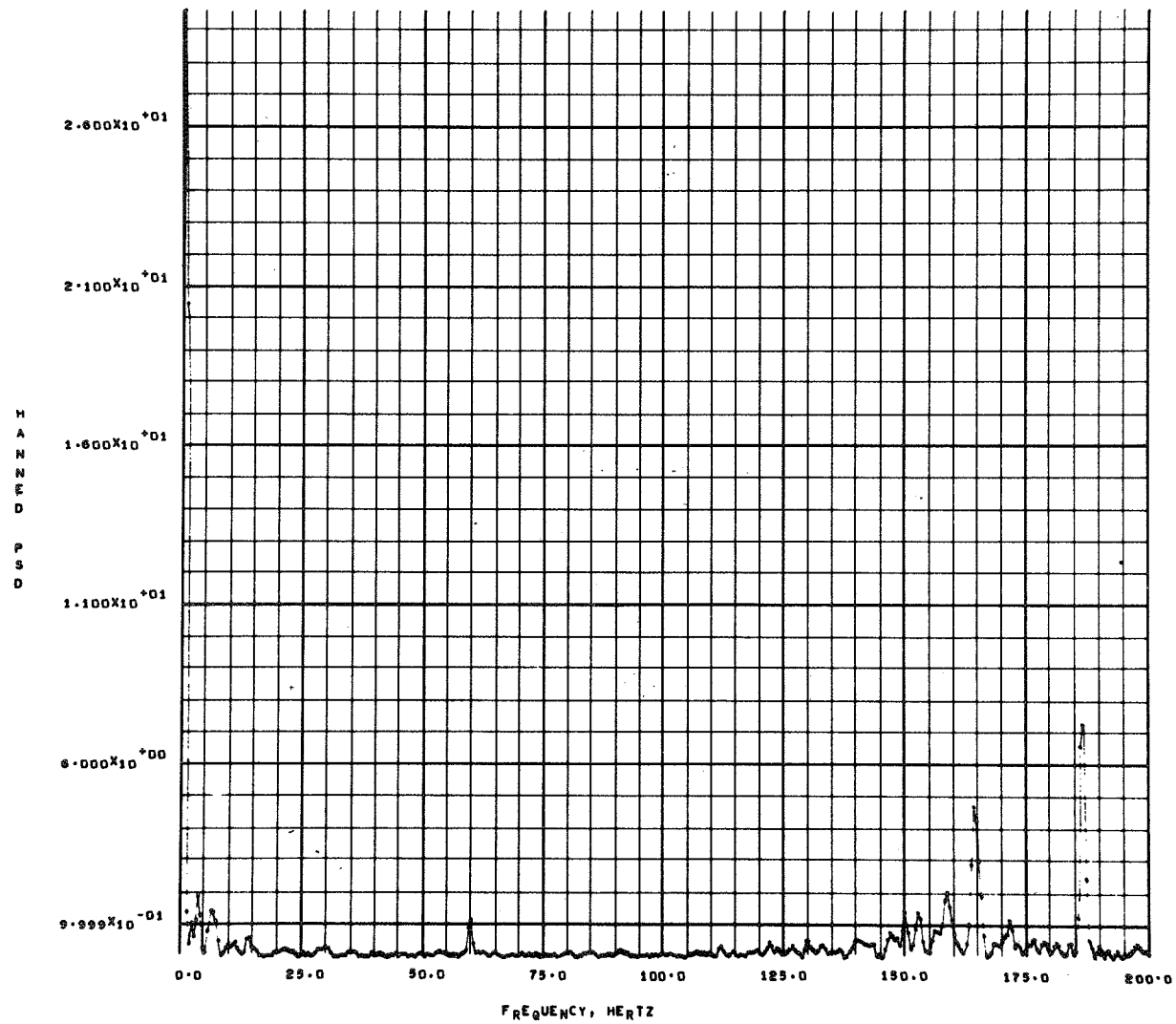


Figure 6. STAN — Power spectral density, Channel 1, 1000 samples per second.

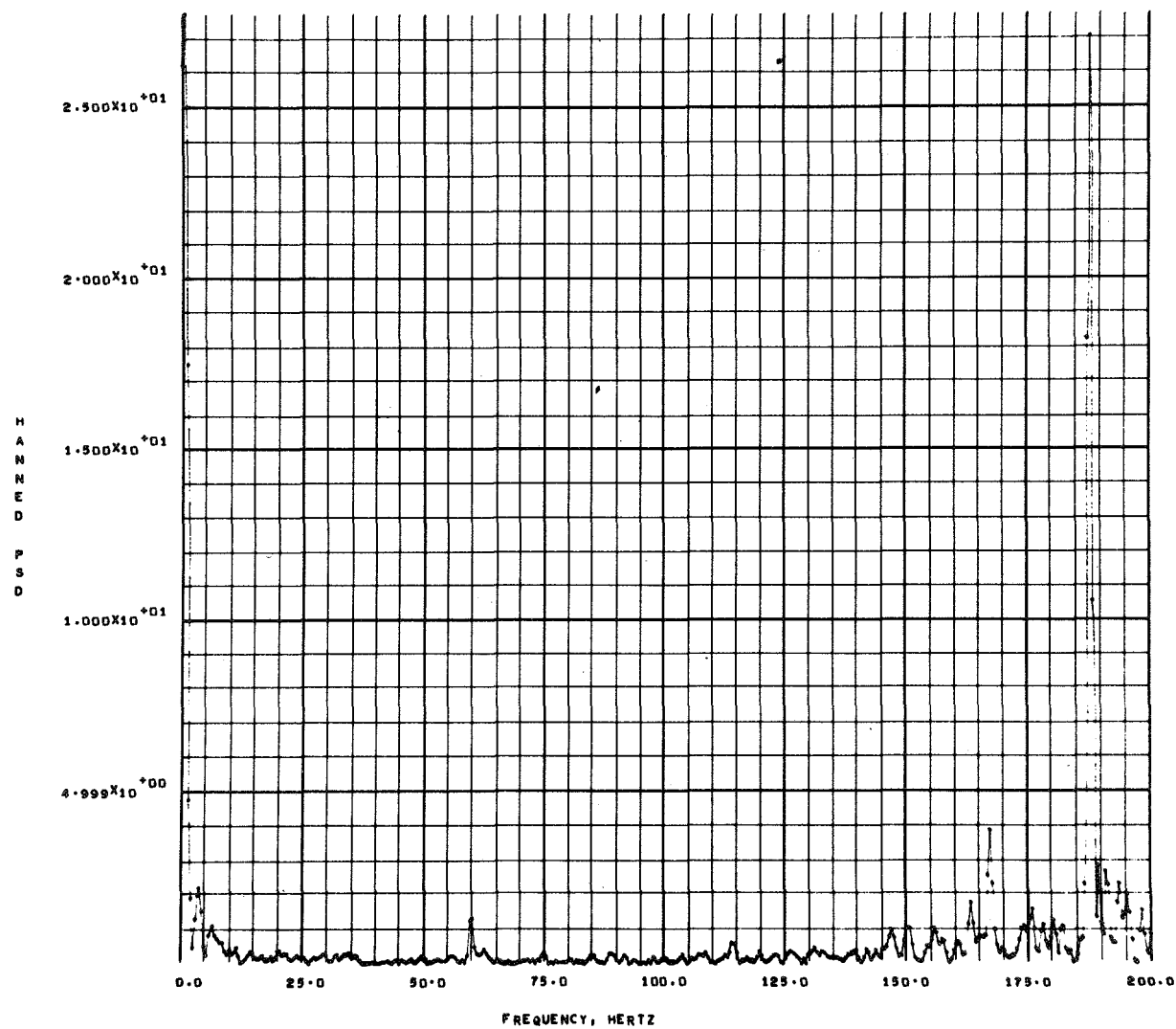


Figure 7. STAN — Power spectral density, Channel 2, 1000 samples per second.

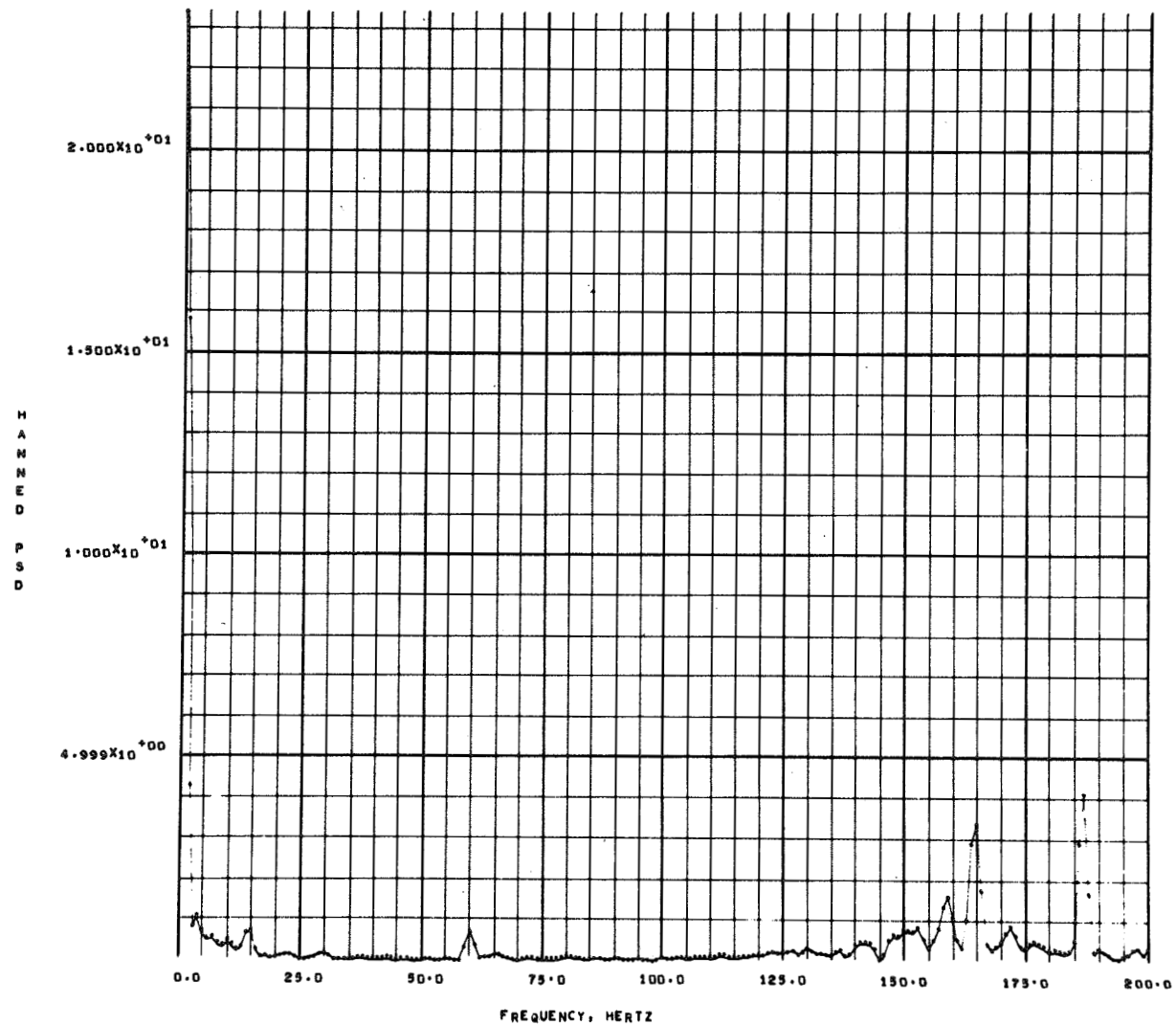


Figure 8. STAN — Power spectral density, Channel 1, 1024 samples per second.

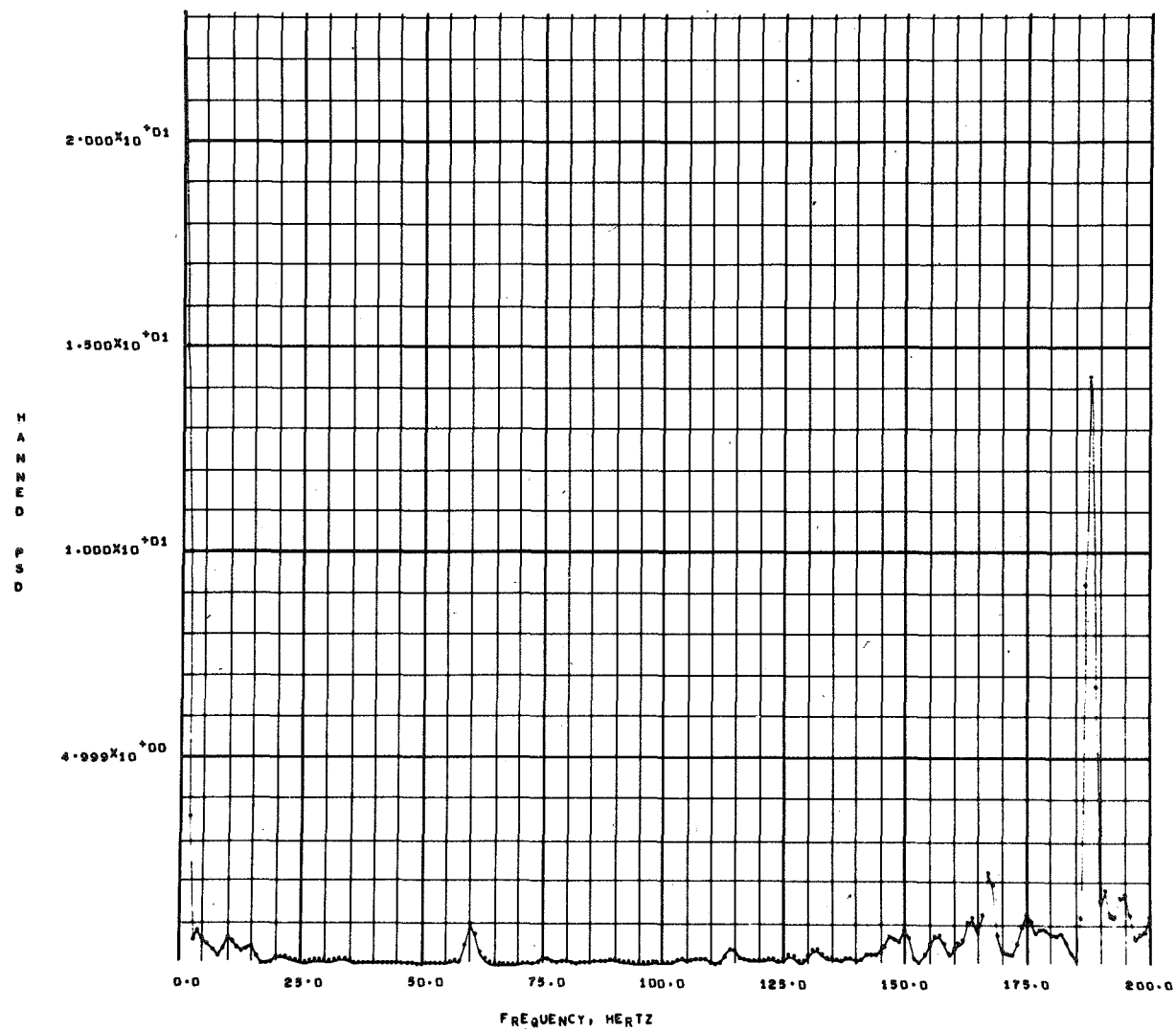


Figure 9. STAN — Power spectral density, Channel 2, 1024 samples per second.

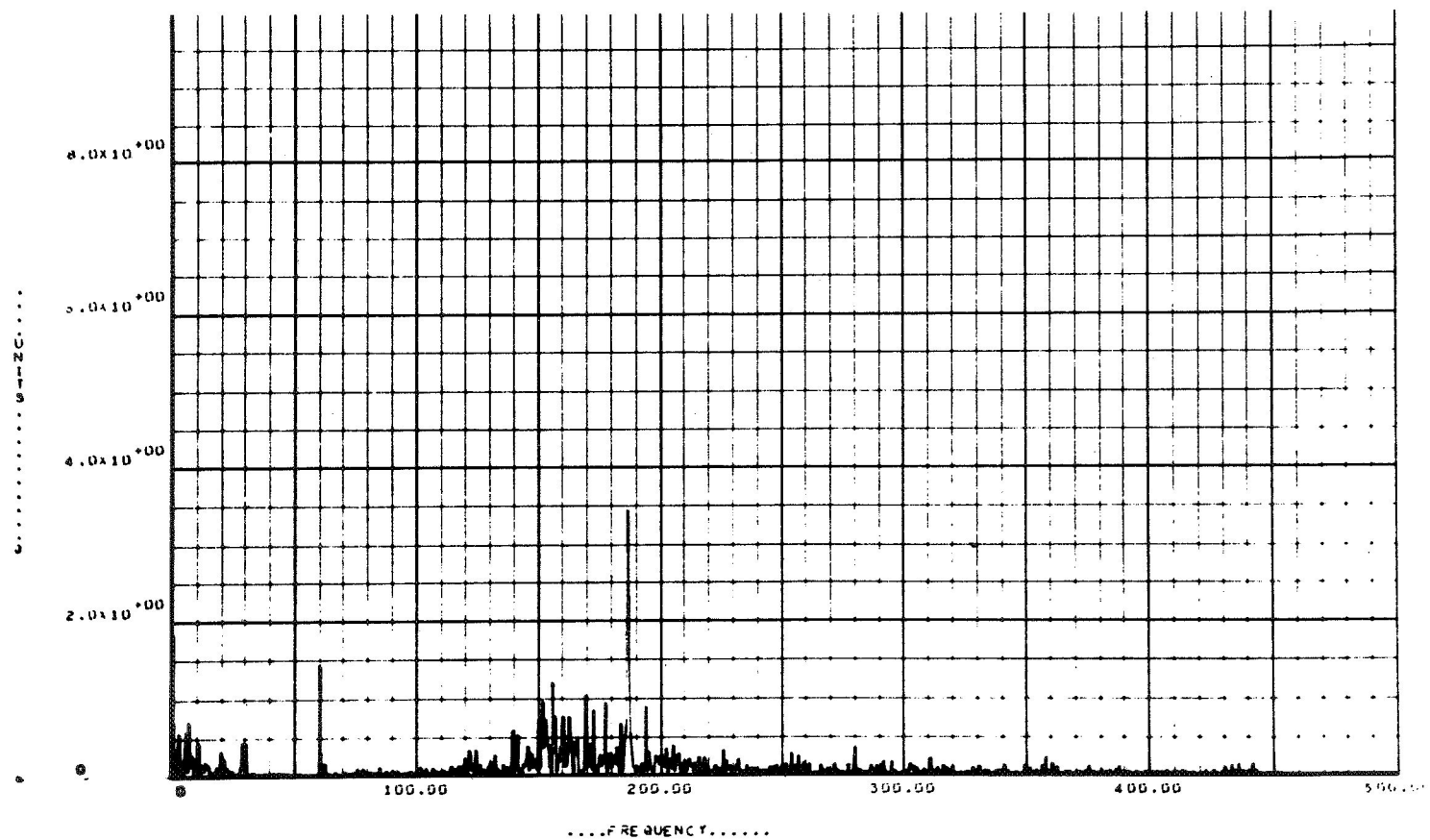


Figure 10. Power spectral density from six-bit data, Channel 1.

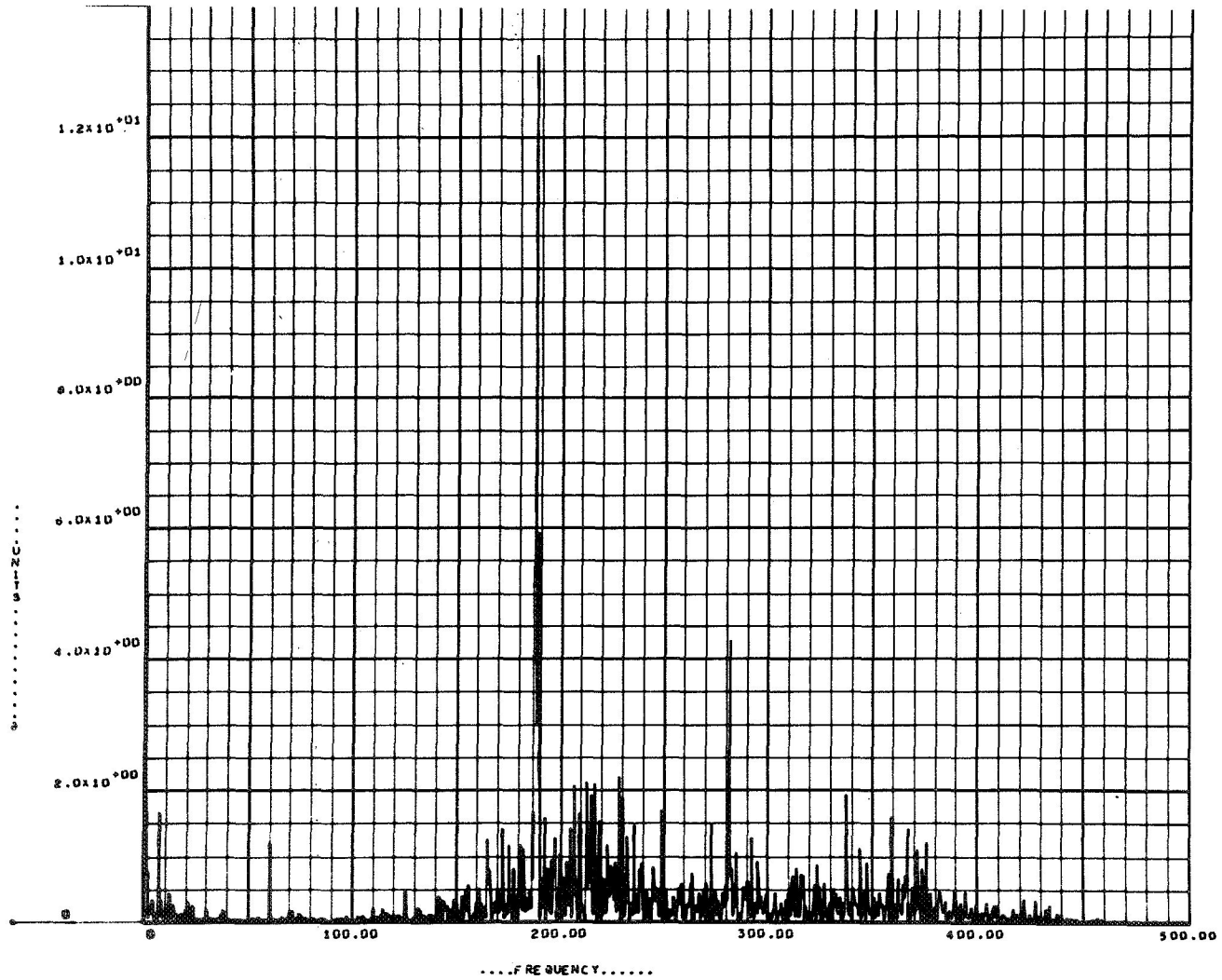


Figure 11. Power spectral density from six-bit data, Channel 2.

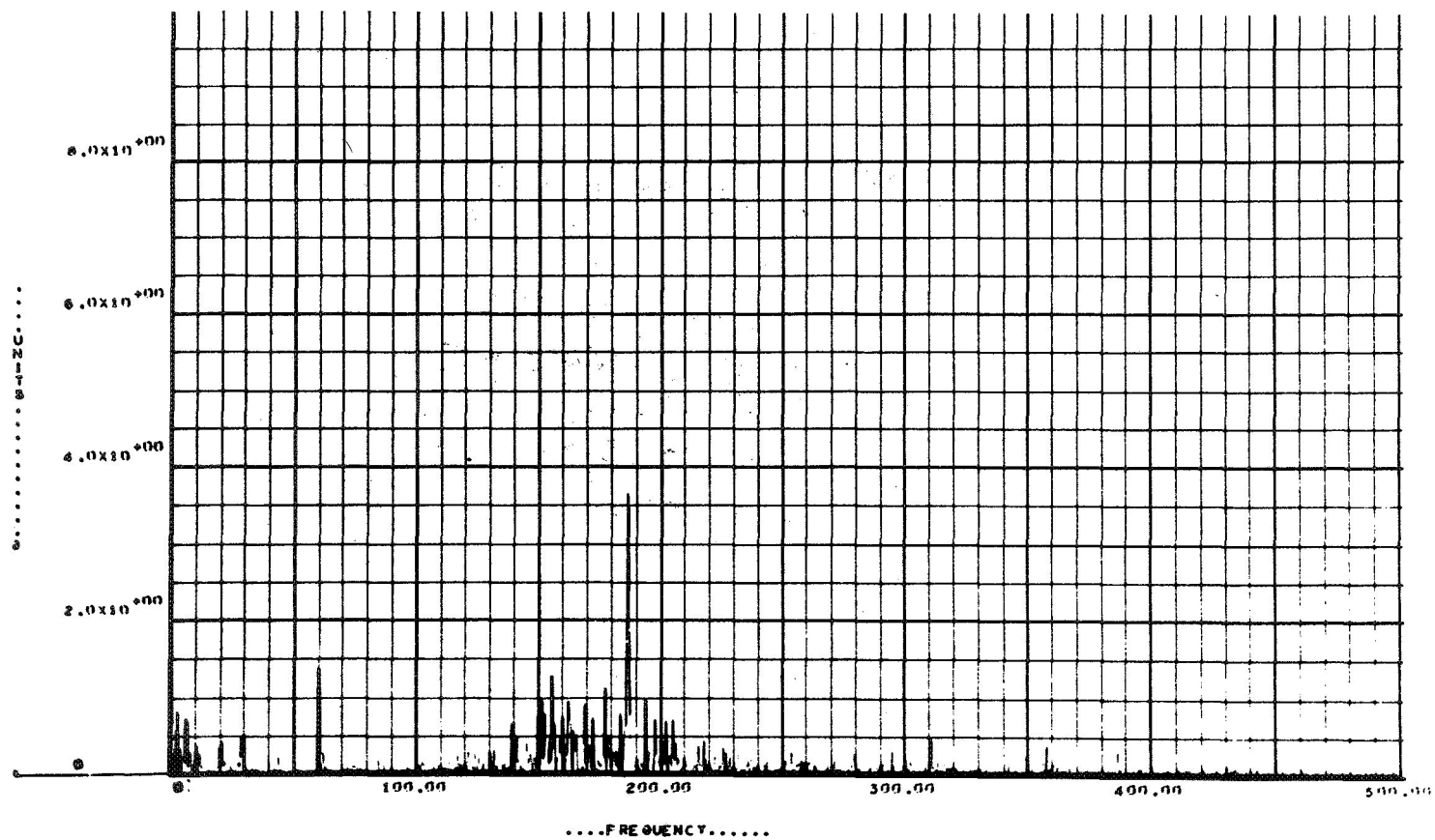


Figure 12. Power spectral density from two-bit data, Channel 1.

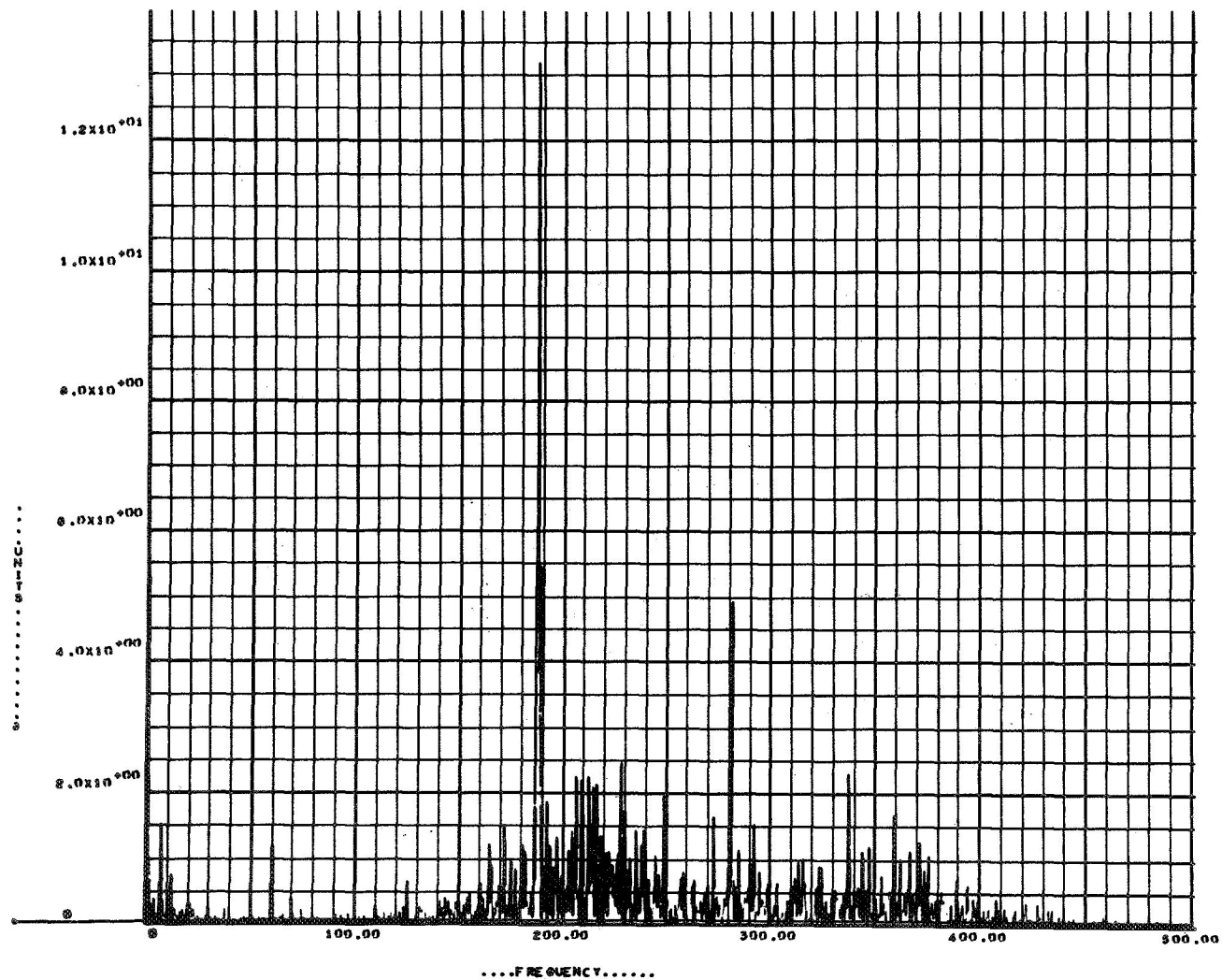


Figure 13. Power spectral density from two-bit data, Channel 2.

The results for these two cases are very accurate. The maximum percentage error for $|H(f)|$ is for the 0 frequency value of $H(f)$ in Table 11 and this is less than 2 percent. The maximum percentage error relative to the maximum value of $|H(f)|$ for the other terms in Table 11 is less than 0.5 percent. In Table 12, the maximum percentage error in $|H(f)|$ relative to the maximum value of $|H(f)|$ is less than 0.2 percent.

INTERPRETATION

The spectral density function is calculated at discrete points. In general, nothing is known about its value between these points. The graphs in Figures 2 through 9 are the products of spectral density function and bandwidth.

Without prior knowledge of the spectral density function, one cannot tell whether indicated energies at adjacent points result from one or more frequency components. In this case, a more detailed analysis is required. Frequently, the analysis of a longer segment of the time series will suffice. This is verified by Table 1.

In the cases studied here, it appears that most of the smooth peaks resulted from one frequency falling between the calculated values. In the two cases of actual observed data, it was obvious that small changes in the points where the FFT was calculated made large changes in the maximum values calculated. In other words, the maximum calculated value may not be a very good approximation of the true maximum value in the neighborhood of that point. However, with the recommended sampling rates and length of series analyzed, the values of the spectral density function are as good approximations of the true values as any other discrete method.

TYPICAL HARDWARE AND SOFTWARE CHARACTERISTICS

The time required to calculate the FFT and the power spectral density depends on the number of values of the time series, the number of bits used to express the value of the function, and the computer and display used.

The number of computation operations required to perform the FFT and N data points is proportional to $N \log_2 N$ versus N^2 for the classical Blackman-Tuckey approach. When N is greater than 1023, the time can be about 100 times faster for the FFT approach than for the classical approach. An even greater time savings can be achieved utilizing hardware to calculate the FFT, which is called the Fast Fourier Analyzer (FFA).

Special purpose hardware, such as the FFA hardware, can be attached to the mainframe computer as a peripheral to one of the mainframe input/output (I/O) channels. The mainframe computer controls the FFA hardware via priority interrupts or other program logic. The FFA hardware has a separate core memory to allow simultaneous operations to be performed by the mainframe computer and FFA hardware. This is a very important advantage for many real-time data processing applications such as image enhancement, spectral analysis, radar, sonar, and vibration analysis. See summary Table 15 for some typical FFA's.

Since the FFA typically produces a squared quantity summed over a large data population N , the number of bits to resolve the correct answer is defined by the following equation:

$$R = J + 2(K) + 1$$

where J is the number of bits that will define the maximum population N to be considered by the FFA. K is the A/D bit word size not including the sign. Therefore, the FFA hardware specifications that define the complex arithmetic bit register resolution to maintain accuracy without data compression is R bits.

Often, the test engineer wants to know how many data channels may be analyzed by the FFA in real time. Since this will depend on the specific FFA and computer used, these timing equations are given as a guide to aid the engineer in this evaluation.

INPUT	$T_i = HBN + P_i$	$\mu \text{ sec}$
OUTPUT	$T_o = HBN + P_s$	$\mu \text{ sec}$
FFT	$T_f = 4N \log N + 2N + P_s$	$\mu \text{ sec}$

$$\text{MULT} \quad T_m = 6N + P_r \quad \mu \text{ sec}$$

$$\text{Spectral Density} \quad T_{SD} = T_f + T_m$$

N is the number of data points (must be a power of two) of the real or imaginary data array;

B is the number of bytes in each element of the data array;

P_i is the program CPU overhead to acquire input data, convert to engineering unit, and load FFA data memory;

P_s is the CPU program overhead to service the priority interrupt service subroutine, store complex results, and decrement counters and other program logic to orderly proceed through the FFT;

H is the channel transfer time per byte;

P_r is the CPU program overhead to convert the FFT real and imaginary array to power spectral densities, and conversion for output display.

Tables 13 and 14 show a typical test case using two different computers to perform the FFT with and without FFA. To perform on-line quick-look analysis, the most time consuming part is the integration time of the FFA (Table 14). In our investigation, the next time consuming item was the display time of the CRT. This limits the number of the data channels that may be analyzed during real time to one data channel every 13 seconds for the SDS-930 computer system and one channel for every five seconds for the SIGMA-5 computer; real time could not be considered for most vibration applications because the software speed of the FFT would be too slow. Without the FFA, the time required for analysis and display for one channel is 98 seconds with the SDS-930 computer and 12.2 seconds for the SIGMA-5 computer. The type of mainframe computer used will influence the choice of the FFA hardware. Also, a fast and more advanced computer may not require FFA hardware for some applications.

TABLE 13. TYPICAL CHARACTERISTICS OF ENVIRONMENTAL TESTS

Type Of Test	Typical Time Duration/Test (sec)	Highest Frequency Content/Ch. (Hz)	Typical Sampling Rate Channel ^a	No Channels Required Total/On-Line	System Recording/Thoroughput Rate (On-Line) ^a
Sine Sweep	0 - 540	2000	8 000 - 10 000	14/2	115 200
	540	1000	4 000 - 5 000	28/12	115 200
	540	500	1 500 - 2 500	32/12	80 000
	540	30	100 - 150	32/12	4 800
Sine Dwell	0 - 10	3000	8 000 10 000	14/2	115 200
	10	1000	4 000 5 000	28/12	115 200
	10	500	1 500 2 500	32/12	80 000
	10	30	100 150	32/12	4 800
Random	0 - 10	5000	10 000 12 000	11/2	115 200
	10	2000	4 000 6 000	28/2	115 200

a. Data samples/second

TABLE 14. FFT TIMING ESTIMATES

Operation		SDS-930		XDS-SIGMA 5	
		N = 2048 (sec)	N = 4096 (sec)	N = 2048 (sec)	N = 4096 (sec)
Acquire $x(t)$ in CPU mainframe memory	T_i	0.5	1.0	0.1	0.2
Transform to engineering units	T_i	0.25	0.5	0.1	0.2
Transform to zero mean	T_i	0.1	0.2	0.05	0.1
Spectra Computation by software without FFA ^a	T_{fs}	90	270	10	31
Load FFA ^b memory	T_i	0.006	0.012	0.003	0.006
FFA solution time (FFT service subroutine plus overhead and spectra conversion) ^b	P_s	5	15	2.5	6.5
CRT display overhead	P_r	7	14	2	4
Total time with FFA simultaneous input/ output throughput ^b		12	27.77	4.753	11.0006
Total time simultaneous input/ output with software FFT ^a		97.856	285.712	12.253	35.506

a. Software option

b. Hardware option

TABLE 15. TYPICAL FFA'S AND THEIR BASIC CHARACTERISTICS

CHARACTERISTICS	BELL LABS	IBM	TEXAS INSTR II VERSION II	TEXAS INSTR III VERSION III	TIME DATA VERSION 1	TIME DATA VERSION 2	WESTING -HOUSE
MAXIMUM N PROCESSED	8192	32,768	4096	4096	1001	2048	2048
MAXIMUM CORE USED	8192	32 K	16384	16384	4096	2048	2048
ALGORITHMS	4 R.M.	1 R.M	1 R.M.	1 R. M.	2 R.M.	2 R.M.	4R.M.
COOLEY TUKEY	H	HS	H	H	OTHER	OTHER	H
STAND ALONE			X	X	X	X	
GP COMPUTER ATTACHMENT	X	X	X		X	X	X
RADIX	2	2/4	2	2			2
PRECISIONS BITS	12	32	18	16	15	12/23	10
BITS INPUT	12	32	8	16	8	12	10
BITS OUTPUT	12	32	18	16	16	24	10
TIMING N=1024	31	56	9	9	25	25	4
EST. COST	\$70,000	\$356,550	NA	\$70,000	\$66,925	\$45,000	N/A

S = SOFTWARE, H = HARDWARE, R.M. = REAL MULTIPLIER, MS = MILLISECONDS.

PROBLEM AREAS

Some work was done to determine the effect of the number of bits used in the quantization of the data on the calculated values of the FFT and the power spectral density function. The few cases studied seem to indicate that the number of bits used was not too significant. Some improvement in the speed of calculation usually follows when fewer bits are used. However, scaling problems are introduced and the effect of the small changes in FFT and power spectra may cause large changes in the autocorrelation function and other related functions. This area needs more study.

Another area of concern is the autocorrelation and cross-correlation functions. Most of the literature recommends some type of filter of the original data. With deterministic data, the results were good without these filters. Study of the effect of these filters on other types of data is indicated.

The transfer function and phase angle need to be investigated further. The use of these functions without separation of frequencies seems to give almost meaningless results.

Figures 14 and 15 show the raw data that were analyzed.

CONCLUSION

The FFT that has been defined and used in this report does yield good estimates in the frequency domain. These estimates are given at discrete intervals (Δf). The FFT, as all other digital methods such as STAN or RAVAN, does not define energy between two adjacent frequency intervals. Values between adjacent points can be interpolated; however, caution has to be taken in the interpretation of these interpolated results. When more resolution is desired, a longer data record can generally be used to obtain this resolution. The FFT does yield results at least equal to previously used techniques such as RAVAN and STAN. More important, the speed advantage of the FFT is several magnitudes greater than the Blackman-Tuckey techniques used by RAVAN. The most important feature of the FFT is that computational algorithm hardware and core memory (FFA) can be added to existing computers to perform rapid real-time signal or data processing. In our investigation, the number of bits that

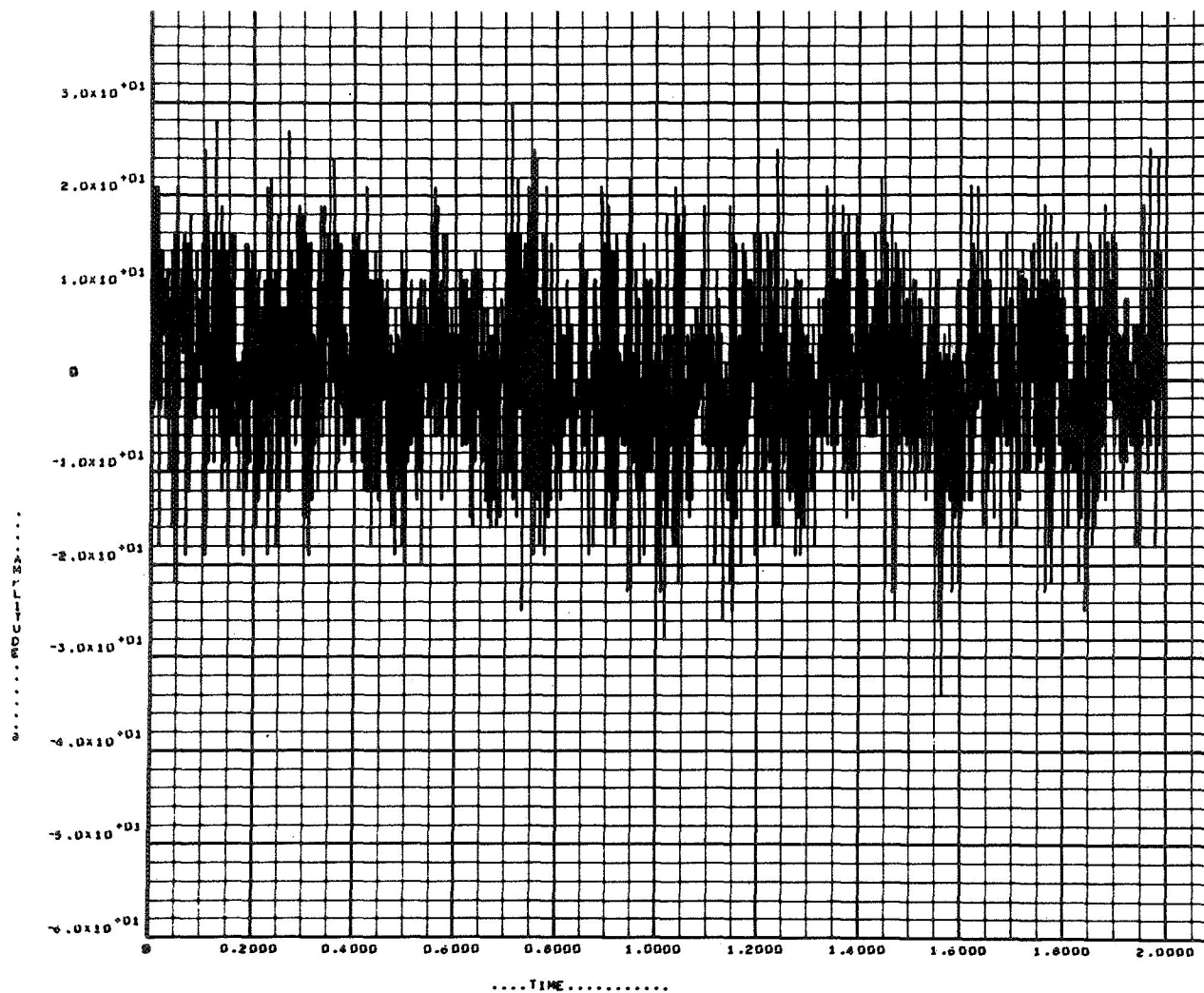


Figure 14. Raw data, Channel 1.

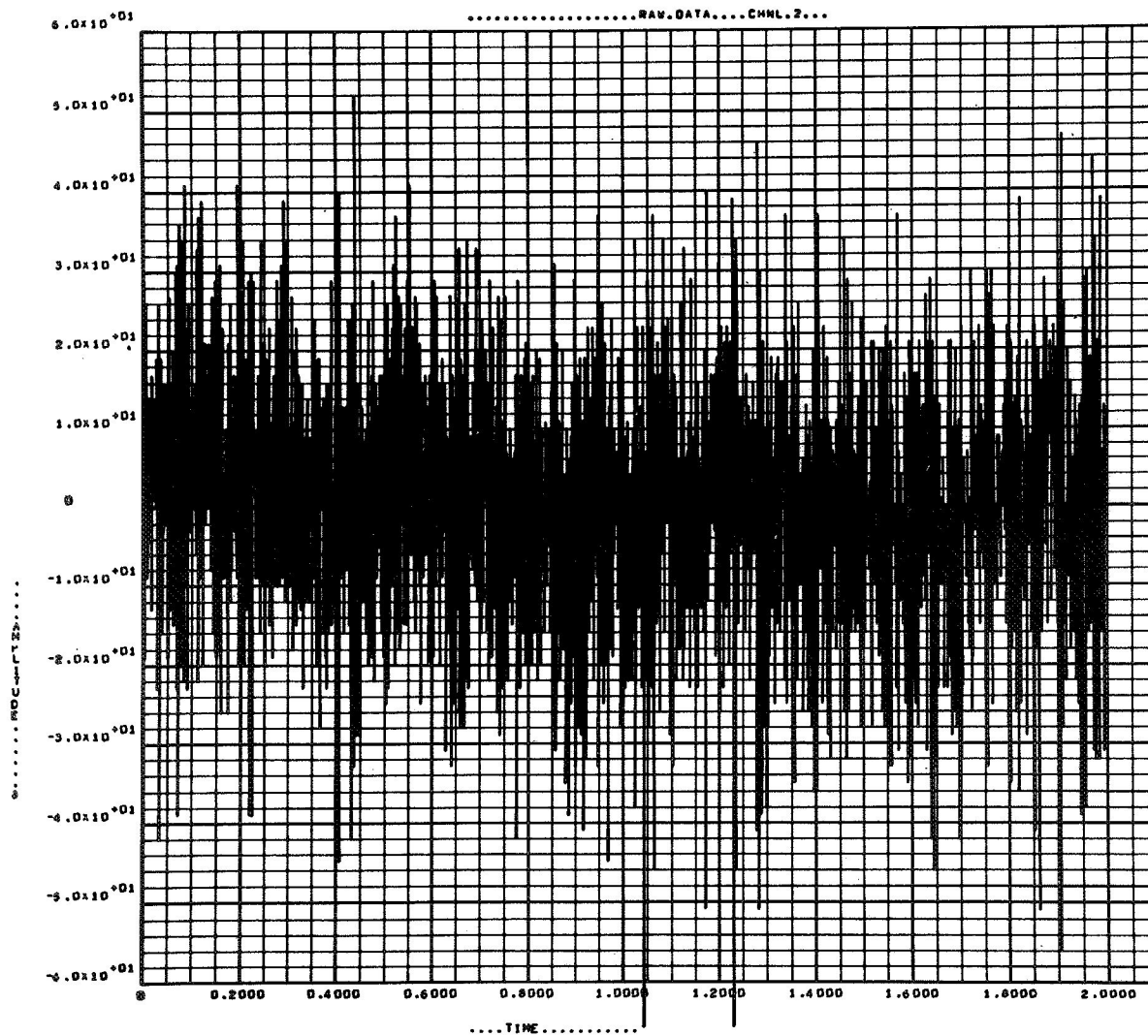


Figure 15. Raw data, Channel 2.

defined each data sample was not as critical as the sampling rate. Figures 10 and 11 showed that reducing the A/D size from 11 bits to 6 bits only reduced the power spectral density (PSD) magnitude of the major five predominant frequencies by 10 percent. When the A/D size was reduced from 11 bits to 2 bits, Figures 12 and 13, the PSD magnitude of the predominant frequency was degraded about 10 percent. As expected, the number of predominant frequencies that are detectable are fewer and the lower amplitudes are degraded the most when two bits are used.

From the foregoing studies it can be concluded that quick-look vibrational analysis of a few channels with a digital computer is feasible for structural testing. A hardware Fast Fourier Analyzer (FFA) connected to the computer as a peripheral reduces considerably the analysis time, particularly in case a relatively slow computer is used. It is expected that with a modern medium size digital computer with an FFA, a complete quick-look vibrational analysis of a signal can be computed and displayed every 5 seconds. If one assumes that a waiting time of 1 minute is acceptable, 12 channels could be monitored simultaneously for vibrational analysis of structures. The study also showed that the time to display the data on the CRT must not be neglected, and that proper selection of the display unit has to be considered.

The FFT was used to calculate the total power in two response functions (time function). These two functions were $100 e^{-10t}$ and $100 e^{-2t} \sin 12\pi t$. These were chosen to represent two realizable systems. The calculated values and the theoretical values were in close agreement. The results are summarized in Tables 11 and 12.

APPENDIX

FAST FOURIER TRANSFORM

The spectral density functions can be defined directly by taking Fourier transforms [1]. There are certain theoretical requirements necessary for the existence of these transforms; they will be assumed to be met. For those interested, there are many rigorous treatments of the Fourier transform. The spectral decomposition of a time series $x(t)$ will be developed by assuming that it has a complex Fourier transform $X(f)$ such that

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

and conversely

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$$

A sufficient condition for the existence of these integrals is that $x(t)$ and its derivative $\dot{x}(t)$ be piecewise continuous in every finite interval (a, b) and the $x(t)$ be integrable on $(-\infty, \infty)$. These conditions can be satisfied (and usually are) in practical problems by setting $x(t)$ equal to zero outside some fixed range of t [that is, making $x(t)$ a finite time series].

Similarly, if there exists a second time function $y(t)$, it will be assumed to have a complex Fourier transform $Y(f)$ such that

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{-j2\pi ft} dt$$

and

$$y(t) = \int_{-\infty}^{\infty} Y(f) e^{j2\pi ft} df$$

In practice, $x(t)$ and $y(t)$ are assumed to be zero outside some range of t and that there are no frequencies above some finite frequency F . Hence, the integrals become

$$X(f) = \int_0^T x(t) e^{-j2\pi ft} dt$$

$$x(t) = \int_{-F}^F X(f) e^{j2\pi ft} df$$

$$Y(f) = \int_0^T y(t) e^{-j2\pi ft} dt$$

and

$$y(t) = \int_{-F}^F Y(f) e^{j2\pi ft} df$$

Also, in practical problems the time series $x(t)$ must be sampled at a certain increment and the sampled values used. This requires the integrals to be redefined as sums of products. Usually, equal spacing is used for time and $X(f)$ and $Y(f)$ are calculated at equally spaced points of frequency. Using the trigonometric form of $e^{j\theta}$, these integrals become

$$X(f_i) = (\Delta t) \sum_{K=0}^{N-1} x(t_k) [\cos 2\pi f_i t_k - j \sin 2\pi f_i t_k]$$

$$x(t_k) = (\Delta f) \sum_{i=0}^{N-1} X(f_i) [\cos 2\pi f_i t_k + j \sin 2\pi f_i t_k]$$

and similarly for $Y(f)$ and $y(t)$.

If $X(f)$ and $Y(f)$ are calculated for a discrete set of f values, they are referred to as Discrete Fourier Transforms or DFT. When the time series $x(t)$ is a discrete set of equally spaced numbers, there is a process of

multiplication that permits faster calculation of $X(f)$. If $t_1 = \frac{T}{N}$ where $N = 2^k$ for some integer and $\Delta f = \frac{1}{T}$, $2\pi f_1 t_1 = 2\pi \left(\frac{1}{T}\right) \left(\frac{T}{N}\right) = \frac{2\pi}{N}$. Then if $e^{-j2\pi f_1 t_1}$ is denoted by α , then $e^{-j2\pi f_k t_1} = \alpha^{kl}$. Using this notation $X(f)$ can be written as

$$\begin{bmatrix} X(0) \\ X(\Delta f) \\ \vdots \\ X[(N-1)\Delta f] \end{bmatrix} = \begin{bmatrix} 1 & \dots & \dots & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{N-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(N-1)} \\ & \vdots & & & \\ 1 & \alpha^{(N-1)} & \dots & \dots & \alpha^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x(t) \\ x(t_1) \\ \vdots \\ x(t_{N-1}) \end{bmatrix}$$

or $X^T(f) = A X^T(t)$, where each value of $X(f)$ must be multiplied by (Δt) for the true value.

If this method is used with the following arithmetic, it is called the Fast Fourier Transform or FFT. The speed of calculation is obtained by factoring A in such a way as to minimize the complex multiplication and addition. It should be remembered that $e^{-2\pi k} = e^0 = 1$ for any integer k and since

$\alpha = e^{-\frac{j2\pi}{N}}$, α^{kl} can be represented as α^t where t is the remainder when kl is divided by N . Using this notation, factoring for $N = 4$ and $N = 8$ is given. If $N = 4$ (i.e., x_0, x_1, x_2, x_3 form the discrete time series), the matrix notation for the Discrete Fourier Transform is

$$X(f) = \Delta t \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} X(0) \\ X(\Delta f) \\ X(2\Delta f) \\ X(3\Delta f) \end{bmatrix}$$

where $\Delta f = \frac{1}{T}$, $\alpha = -\frac{j\pi}{2}$, $\alpha^{4k+1} = \alpha^1$ and $l = 0, 1, 2, 3$.

The Fast Fourier Transform factors A as follows:

$$Q = \begin{bmatrix} 1 & \alpha^4 & 0 & 0 \\ 1 & \alpha^2 & 0 & 0 \\ 0 & 0 & 1 & \alpha^4 \\ 0 & 0 & 1 & \alpha^2 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & \alpha^2 & 0 \\ 0 & \alpha & 0 & \alpha^3 \end{bmatrix},$$

but

$$QR = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 \end{bmatrix}$$

The two middle rows have to be interchanged. Hence, if one does the matrix multiplication $y^T(t) = R[x(t)]$, then $y^T(t) = Qy^T(t)$. The rows must be decoded. Using binary subscripts and starting with 0,

$$\begin{aligned} R_{00} &\rightarrow R_{00} \\ R_{01} &\rightarrow R_{10} \\ R_{10} &\rightarrow R_{01} \\ R_{11} &\rightarrow R_{11} \end{aligned} .$$

In other words the second element of Z^T would be the third element of AX^T and the third element of Z would be the second element of AX^T . The first and fourth (zero and third) would be in the same locations.

If $N = 8$ (i.e., $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7$ form the discrete time series), the matrix multiplication for the Discrete Fourier Transform is

$$X^T(f_i) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & . & . & . & \alpha^7 \\ 1 & \alpha^2 & \alpha^4 & . & . & . & . & \alpha^{14} \\ 1 & \alpha^3 & . & . & . & . & . & \alpha^{21} \\ 1 & \alpha^4 & . & . & . & . & . & \alpha^{28} \\ 1 & \alpha^5 & . & . & . & . & . & \alpha^{35} \\ 1 & \alpha^6 & . & . & . & . & . & \alpha^{42} \\ 1 & \alpha^7 & . & . & . & . & . & \alpha^{49} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ . \\ . \\ . \\ x_7 \end{bmatrix}$$

and $X[(k-1)\Delta f]$ is the vector dot product of the k th row of the matrix and the column of x values. The powers of α are not reduced module 8 for reasons

that will become apparent $\left(\alpha = e^{-\frac{j2\pi}{8}} \right)$. This same matrix is used for the Fast Fourier Transform but is factored into three matrices and multiplied in that way.

The matrix factorization is done in the following way. Take the first two rows ($2 \times 4 = 8$) and divide each one into two equal parts. The first half of the first row is the diagonal of a 4×4 submatrix in the upper left corner. The second half of the first row is the diagonal of a 4×4 matrix in the upper right corner. Then the first half of the second row is the diagonal of a 4×4 matrix in the lower left corner and the remainder of the second row is the diagonal of a 4×4 submatrix in the lower right. In short, one of the factors is

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & \alpha^4 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & \alpha^5 & 0 & 0 \\ 0 & 0 & \alpha^2 & 0 & 0 & 0 & \alpha^6 & 0 \\ 0 & 0 & 0 & \alpha^3 & 0 & 0 & 0 & \alpha^7 \end{bmatrix}$$

Now, the other factor must be found. It is

$$Q = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & 0 & 0 & 0 & 0 \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} & 0 & 0 & 0 & 0 \\ 1 & \alpha^6 & \alpha^{12} & \alpha^{18} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 0 & 0 & 0 & 0 & 1 & \alpha^4 & \alpha^8 & \alpha^{12} \\ 0 & 0 & 0 & 0 & 1 & \alpha^6 & \alpha^{12} & \alpha^{18} \end{bmatrix}$$

However QR interchanges rows of A so that they occur $R_1, R_3, R_5, R_7, R_2, R_4, R_6, R_8$. (Perhaps one would rather say $R_0, R_2, R_4, R_6, R_1, R_3, R_5, R_7$, since if binary subscripts are used this is more consistent.)

Now, Q must be factored. If these are designated as $Q_1 R_1$,

$$R_1 = \left[\begin{array}{cccc|c} 1 & 0 & 1 & 0 & \\ 0 & 1 & 0 & 1 & \\ 1 & 0 & \alpha^4 & 0 & \\ 0 & \alpha^2 & 0 & \alpha^6 & \\ \hline & 0 & & & B \end{array} \right],$$

where B is the matrix in the upper left hand corner. Basically the same procedure as used to get R from A is used on each 4×4 matrix on the diagonal of Q. Then, Q is a matrix with 2×2 matrices on the diagonal so that

$$Q_1 = \begin{bmatrix} 1 & \alpha^8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \alpha^4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \alpha^8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \alpha^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha^8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \alpha^4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha^8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha^4 \end{bmatrix}$$

Actually, one should remember that $\alpha^8 = 1$, $\alpha^4 = -1$ and in general $\alpha^{(8k+n)} = \alpha^n$ where $n = 0, 1, \dots, 7$; also (Q, R) interchanges rows of Q . The old rows of Q are now ordered $R_1', R_3', R_2', R_4', R_5', R_7', R_6', R_8'$, where R_k' was the k th row in Q . Hence, the rows are now ordered quite differently from the original matrix A . In matrix notation, the relationship can be written $A = PQR = P(P_1 Q_1 R_1) R$, where P is a matrix obtained from the identity matrix by doing to it what must be done to R to give A ; that is, first row is left alone, second row becomes third, third row becomes fifth, fourth row becomes seventh, fifth row becomes second, sixth row becomes fourth, and seventh row becomes sixth. This last row is unchanged. This gives

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, similarly

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$P P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In other terms the first and eighth rows of $Q_1 R_1 R$ are the first and eighth rows of A , the fifth row of the product is the second row of A , etc. An easier way to remember is to use binary subscripts on the rows and start with 0. Then they are R_{000} , R_{001} , R_{010} , R_{011} , R_{100} , R_{101} , R_{110} , R_{111} . Reverse the order of the ones and zeros. The rows in the product $Q_1 R_1 R$ must be reordered in that manner; that is,

<u>In A</u>		<u>In Product</u>
0 — R_{000}	\longleftrightarrow	R_{000}
R_{001}	\longleftrightarrow	R_{100}
R_{010}	\longleftrightarrow	R_{010}
R_{011}	\longleftrightarrow	R_{110}
R_{100}	\longleftrightarrow	R_{001}
R_{101}	\longleftrightarrow	R_{101}
R_{110}	\longleftrightarrow	R_{011}
R_{111}	\longleftrightarrow	R_{111}

For clarity the fourth row starting with the 0 row of $Q_1 R_1 R$ is the first row of A .

The order of multiplication then becomes

$$Z^T = R X^T$$

$$W^T = R_1 Z^T$$

$$U^T = Q_1 W^T$$

and

$$X^T(f) = P P_1 U^T$$

or $X^T(f)$ is U^T decoded by the process explained above. The factor (Δt) is usually omitted until the last step.

There are several advantages using this method. One of these is as follows. $X(f)$ is permitted to be complex. If one has two real time series, let $Z(t_i) = x(t_i) + j y(t_i)$. Find the FFT of $z(t)$ or $Z(f)$. Then

$$X(k\Delta f) = \frac{Z(k\Delta f) + \bar{Z}[(N-k)\Delta f]}{2}$$

and

$$Y(k\Delta f) = \frac{Z(k\Delta f) - \bar{Z}[(N-k)\Delta f]}{2j},$$

where \bar{Z} is the complex conjugate of Z .

Another special application is the following. If $x(t)$ is a discrete (equally spaced) time series of length 2^k , let

$$x_1(t) = (x_0, x_2, x_4, \dots, x_{N-2})$$

$$x_2(t) = (x_1, x_3, x_5, \dots, x_{N-1})$$

be the two discrete time series of length 2^{k-1} formed as shown. Then enter $x_1(t)$ as the real part of $x(t)$ and $x_2(t)$ as the imaginary part of $x(t)$ in the matrix. Now, noting that the time spacing in $x_1(t)$ and $x_2(t)$ is $2\Delta t$, the FFT of $x(t)$ can be obtained by one pass through FFT process. Let $A_1(f)$ be the transform of $x_1(t)$ and $A_2(f)$ be the transform of $x_2(t)$. Then, for the original series $x(t)$,

$$X(k\Delta f) = \frac{1}{2} \left[A_1(k\Delta f) + A_2(k\Delta f) e^{-\frac{j2\pi k}{N}} \right]$$

for $k < \frac{N}{2}$ and

$$X[(N-k)\Delta f] = \overline{X(k\Delta f)}$$

for $(N-k) > \frac{N}{2}$.

Since the correlation function is the inverse Fourier transform of the power spectral density function, the FFT routine may be used to calculate the correlation function. Actually, the entries in the matrix notation would have positive powers. However, the problem is easily solved by complex arithmetic. Most programs use $e^{j2\pi ft}$ rather than $e^{-j2\pi ft}$ as used in this report. If this is the case, then the actual Fourier transform is the complex conjugate of the calculated FFT. If this is the case, then using $S_x(f)$ as the entry for $x(t)$ yields the autocorrelation function. In matrix notation, using $C_{xx}(t)$ for the correlation function,

$$\begin{bmatrix} C_{xx}(0) \\ C_{xx}(t_1) \\ \vdots \\ C_{xx}(t_{N-1}) \end{bmatrix} = (\Delta f) \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{+ \frac{j2\pi}{N}} & e^{\frac{j4\pi}{N}} & e^{\frac{j(N-1)2\pi}{N}} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & e^{\frac{(N-1)j2\pi}{N}} & e^{\frac{j(N-1)^2 2\pi}{N}} & \end{bmatrix} \begin{bmatrix} X(0) \\ X(\Delta f) \\ \vdots \\ X[(N-1)\Delta f] \end{bmatrix}$$

or

$$C_{xx}^T(t) = B X^T(f),$$

where each element of B is the complex conjugate of the matrix A in the calculation of the FFT. In most subroutines, the matrix B , given above is the one used. Logic steps easily change the calculated values into the correct Fourier transform.

REFERENCES

1. Enochson, L. D. and Piersol, A. G.: Application of Fast Fourier Transform Procedures to Shock and Vibration, Data Analysis. Report No. 670874, Society of Automotive Engineers, October 2-6, 1967, p. 10.
2. Bingham, C., Godfrey, M. D., and Tuckey, J. W.: Modern Techniques of Power Spectral Estimation. Vol. AU-15, IEEE Trans. Audio and Electroacoustics, June 1967, pp. 56-66.
3. Sloane, E. A.: Comparison of Linearly and Quadratically Modified Spectral Estimates of Gaussian Signals. Vol. AU-17, IEEE Trans. Audio and Electroacoustics, June 1969, pp. 133-137.
4. Newberry, Murl H.: Random Vibration Analysis Program (RAVAN). NASA Technical Memorandum X-53359, November 1965.
5. Barber, N. J.: Experimental Correlograms and Fourier Transforms — International Tracts in Computer Science and Technology and Their Application. Vol. 5, Pergamon Press, 1961.
6. Larson, A. G. and Singleton, R. C.: Real-Time Spectral Analysis on a Small General-Purpose Computer. AFIPS Proceedings, Vol. 31, 1967 Fall Joint Computer Conference, pp. 665-674.
7. Milne, W. E.: Numerical Calculus. Princeton University Press, Princeton, N. J., 1949.
8. Welch, P. D.: A Fixed-Point Fast Fourier Transform Error Analysis. Vol. AU-17, IEEE Trans. Audio and Electroacoustics, June 1969, pp. 151-157.

BIBLIOGRAPHY

Bergland, G. D.: Fast Fourier Transform Hardware Implementations — An Overview. Fast Fourier Transform Hardware Implementation — A Survey. Vol. AU-17, IEEE Trans. Audio and Electroacoustics, June 1969, pp. 104-119.

Cooley, J. W., Lewis, P. A., and Welch, P. D.: The Finite Fourier Transform. Vol. AU-17, IEEE Trans. Audio and Electroacoustics, June 1969, pp. 77-86.

Singleton, R. C.: An Algorithm for Computing the Mixed Radix Fast Fourier Transform. Vol. AU-17, IEEE Trans. Audio and Electroacoustics, June 1969, pp. 93-103.

Welch, P. D.: The Use of the Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short Modified Periodograms. Vol. AU-15, IEEE Trans. Audio and Electroacoustics, June 1967, pp. 70-73.

SOME APPLICATIONS AND LIMITATIONS OF THE FAST FOURIER TRANSFORM

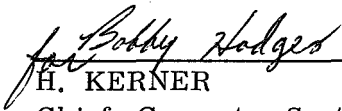
By Edgar Hopper and Murl Newberry

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

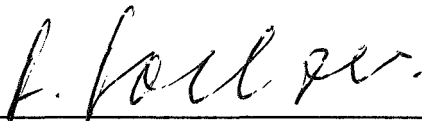
This document has also been reviewed and approved for technical accuracy.



H. TRAUBOTH
Chief, Systems Analysis Branch



H. KERNER
Chief, Computer Systems Division



H. HOELZER
Director, Computation Laboratory

DISTRIBUTION

TM X-53997

INTERNAL

DIR

DEP-T

AD-S

A&TS-PAT

Mr. L. D. Wofford, Jr.

PM-PR-M

A&TS-MS-H

A&TS-MS-IP (2)

A&TS-MS-IL (8)

A&TS-TU (6)

S&E-COMP-C

Hopper (75)

Scientific and Technical Information Facility (25)

P. O. Box 33

College Park, Maryland 20740

Attn: NASA Representative (S-AK/RKT)